
# Deeply-Integrated Feature Tracking for Embedded Navigation

THESIS

Jeffery R. Gray, First Lieutenant, USAF

AFIT/GE/ENG/09-17

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/09-17

# Deeply-Integrated Feature Tracking for Embedded Navigation

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Jeffery R. Gray, B.S.E.E., B.S.C.E.

First Lieutenant, USAF

March 2009

AFIT/GE/ENG/09-17

# Deeply-Integrated Feature Tracking for Embedded Navigation

Jeffery R. Gray, B.S.E.E., B.S.C.E.
First Lieutenant, USAF

Approved:

| | |
|---|---|
| /signed/ | 10 Mar 2009 |
| Lt Col M.J. Veth, PhD (Chairman) | date |
| /signed/ | 10 Mar 2009 |
| Dr. M. Fickus (Member) | date |
| /signed/ | 10 Mar 2009 |
| Maj M.J. Mendenhall, PhD (Member) | date |

# *Abstract*

The Air Force Institute of Technology (AFIT) is investigating techniques to improve aircraft navigation using low-cost imaging and inertial sensors. Stationary features tracked within the image are used to improve the inertial navigation estimate. These features are tracked using a correspondence search between frames. Previous research investigated aiding these correspondence searches using inertial measurements (i.e., stochastic projection). While this research demonstrated the benefits of further sensor integration, it still relied on robust feature descriptors (e.g., SIFT or SURF) to obtain a reliable correspondence match in the presence of rotation and scale changes. Unfortunately, these robust feature extraction algorithms are computationally intensive and require significant resources for real-time operation. Simpler feature extraction algorithms are much more efficient, but their feature descriptors are not invariant to scale, rotation, or affine warping which limits matching performance during arbitrary motion. This research uses inertial measurements to predict not only the location of the feature in the next image but also the feature descriptor, resulting in robust correspondence matching with low computational overhead.

This novel technique, called deeply-integrated feature tracking, is exercised using real imagery. The term *deep integration* is derived from the fact inertial information is used to aide the image processing. The navigation experiments presented demonstrate the performance of the new algorithm in relation to the previous work. Further experiments also investigate a monocular camera setup necessary for actual flight testing. Results show that the new algorithm is 12 times faster than its predecessor while still producing an accurate trajectory. Thirty-percent more features were initialized using the new tracker over the previous algorithm. However, low-level aiding techniques successfully reduced the number of features initialized indicating a more robust tracking solution through deep integration.

*Acknowledgements*

Foremost, I would like to thank to my advisor for his guidance and diligence throughout this research. While we worked hard every step of the way, you made the process fun and rewarding.

Thank you to my committee, former coworkers, and peers for repeatedly reading and discussing this thesis and the techniques described within. Your battered eyes and ears have resulted in a quality, professional document.

I give great thanks to my family for their love and support throughout my academic career. To my father, mother, and sister: thank you for the words of encouragement and unwavering support. To my brother: thanks for constant counsel and advice. Nothing took me by surprise.

Finally, I would like to thank my loving girlfriend. Though you may not immediately realize it, without you I could never achieve so much. Because of you, any stress always seems to melt away. I'm lucky to have such a special woman in my life - I love you, just the way you are.

Jeffery R. Gray

*Table of Contents*

## *List of Figures*

# *List of Tables*

# *List of Abbreviations*

# Deeply-Integrated Feature Tracking for Embedded Navigation

## I. Introduction

Unmanned aerial systems (UAS) continue to develop and inundate every aspect of warfare. In 2000, the United States Department of Defense owned and operated less than 50 UAS systems. Today more than 6,000 systems find diverse missions throughout the battlespace [27]. These systems carry a wide variety of sensor payloads ranging from intelligence, surveillance, and reconnaissance (ISR) to munitions. Not surprisingly, the Department of Defense seeks to continue expansion into more difficult and dangerous missions with priority on ISR and targeting. In addition, such systems should be nearly or completely autonomous to avoid burdening the warfighter in the execution of their mission [28].

Enclosed areas such as indoors, underground, or urban environments present a challenge and are currently only accessible to soldiers or limited-capability ground systems. In these environments, traditional navigation systems cease to function with the Global Positioning System (GPS) either unavailable or degraded. Non-traditional, relative navigation systems provide the autonomy and accuracy necessary to execute these ISR and engagement missions.

The Air Force Institute of Technology and Air Force Research Laboratory are cooperating to develop airborne vehicles capable of operating freely and autonomously in all environments, including indoor environments. To accomplish this goal, several non-traditional sensors are being fused to deliver autonomous systems not dependent on external reference. This research considers fusion of imaging and inertial sensors.

### *1.1 Problem Statement and Scope*

Toward implementing a fully autonomous indoor flying vehicle using non-traditional navigation, this research leverages a fused imaging and inertial system previously developed at the Air Force Institute of Technology [38]. This algorithm is referred to in this research as the image-aided Kalman filter (IAKF). Image and inertial fusion within the filter occurred using a method of stochastic feature tracking. A feature is a distinct point in the image, and feature tracking refers to the detection, extraction, and matching of features between frames. Feature matching benefitted from the stochastic constraint of the search space. However, the robust feature detection and extraction proposed in previous research requires a high level of computation not currently achievable on small indoor flying platforms. This research develops an alternative to a high-level feature detection and extraction that uses inertial information in matching to achieve similar results. In particular, this research attempts to answer the following two questions:

- Can the IAKF be modified to fit on a small vehicle?
- Can inertial information be used to improve feature tracking at a deeper level, in addition to the previous stochastic correspondence search constraint?

The term deeply-integrated refers to the incorporation of inertial information into the image processing, rather than just a reduction in correspondence search space.

### *1.2 Proposed Solution*

The proposed solution for implementing the IAKF on a small indoor flyer involves finding an alternative to high-level feature tracking. This research asserts that the level of computation used in the high-level feature detection and extraction algorithm is unnecessary when combined with inertial information. The following research development goals were established:

- Modify the IAKF to use low-level feature detection, extraction, and matching.

- Develop additional processing techniques to improve low-level feature matching using inertial information.

The developed algorithm will be validated with a set of indoor flight experiments.

This thesis is organized as follows. Chapter II introduces the background information fundamental to understanding the development of the deeply-integrated feature tracking algorithm. Chapter III discusses the methodology of the produced solution. In Chapter IV, the results of three image experiments are covered. Finally, Chapter V gives a summary of conclusions drawn from the experiments and proposes ideas for further research.

# II. Background

This chapter covers the background information required to develop the deeply-integrated feature tracking algorithm. When possible, notation was adopted from previous research [38].

This chapter begins with the definition of the navigational coordinate frames. Next, the image acquisition system used to aide the inertial system is introduced. The acquisition system includes relevant projection, spatial aliasing, and the camera model theory. Image processing techniques are reviewed next in preparation for discussions about feature detection, extraction, and matching. Motion estimation using vision and Kalman filtering is then covered. The related research in the field of vision-aided inertial systems is reviewed, and the previous research is introduced. Finally, the specifics the image-aided Kalman filter are covered along with details of stochastic feature tracking.

## *2.1 Coordinate Frames*

Coordinate frames are fundamental to the study and analysis of navigational system. The coordinate frames used in this research are as follows [38].

- True inertial frame (*I-frame*)
- Earth-fixed inertial frame (*i-frame*)
- Earth-centered, earth-fixed frame (*e-frame*)
- Navigation frame (*n-frame*)
- Body frame (*b-frame*)
- Camera bar frame ($c_0$*-frame*)
- Camera frame (*c-frame*)

Newton's laws of motion only apply in a truly inertial frame (*I-frame*). This frame does not rotate or translate. There is no true inertial frame in this research.

The Earth-fixed inertial frame (*i-frame*) has its origin at the center of mass of the Earth. The axis of rotation defines the z axis of the system, and the x axis lies in the equatorial plane aligned with the fixed stars. The y axis is defined by the right-hand rule. This frame is only a close approximation of an inertial frame since the earth revolves around the sun. However, this frame is a valid inertial frame for terrestrial navigation.

The Earth-centered Earth-fixed frame (*e-frame*) is defined identically to the *i-frame* except that the frame rotates with the axis of the Earth. The x axis is fixed at the intersection of the Greenwich meridian and the equatorial plane. The z axis is defined along the Earth's axis of rotation, and the y axis is defined according to a right-handed Cartesian system.

The navigation frame (*n-frame*) is a locally defined frame with the origin at the center of the navigational system. A north-east-down (NED) axis alignment is assumed for this research. Down is tangential to the surface of the Earth and points toward the center of the Earth.

The body frame (*b-frame*) is defined with respect to the roll, pitch, and yaw axes of the body. Specifically, the x, y, and z axis point out the nose, out the starboard, and out the bottom on the body. Figure 2.1 shows the the body frame of and aircraft.

The camera frame (*c-frame*) and camera bar frame ($c_0$*-frame*) are identical except for a translation. The x, y, and z axis are defined as up, right, and forward when looking out of the camera lens [10] [36]. The camera bar frame is used as the common line-of-sight reference between the two cameras. Figure 2.2 shows the camera and camera bar frames.

*2.1.1 Direction Cosine Matrix.* A direction cosine matrix (DCM) is used to rotate from one coordinate frame to another. Here, a DCM from the previous from to the new frame is denoted by: $C^{new}_{previous}$. For instance, the $C^{c'}_{c}$ DCM from Section 2.3.5 defines the rotation from the initial camera frame ($c$) to the next camera frame ($c'$).

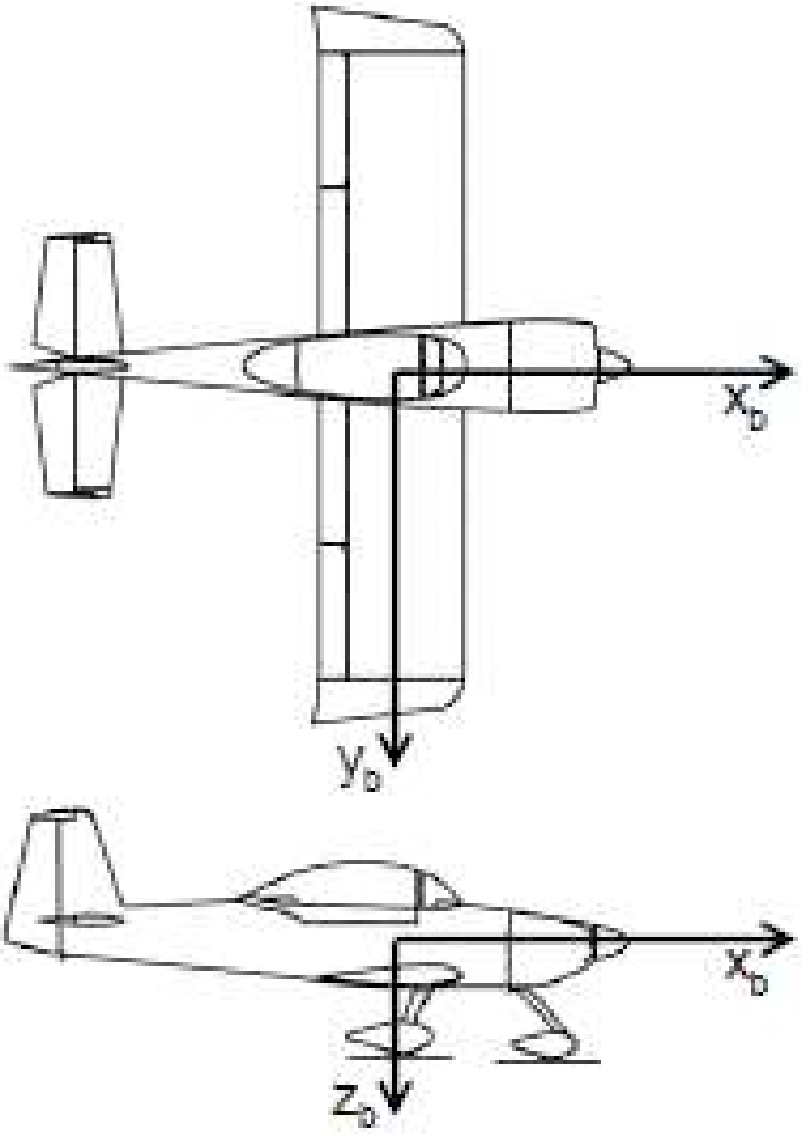



Figure 2.1: Body Frame Diagram [38]. The body frame is defined according to the roll, pitch, and yaw axes of the body.

In the next section, the image acquisition system is introduced and characterized.

## *2.2 Image Acquisition*

Careful analysis of image acquisition is pivotal to proper measurement within a vision system. Because digital cameras are used in this research, this analysis focuses on the optics and charged-coupled device (CCD). Optics focus light onto the image plane where the CCD captures the electric charge of photons delivered over time. The intensity captured is normally quantized on an 8-bit scale [0-255]. The recorded intensity pattern is called the image, and the objects that reflect light into the camera are collectively referred to as the scene. The following sections introduce key concepts of image acquisition important to this research.

*2.2.1 Radiometry.* Radiance is the amount of energy emitted from an object, and irradiance is the amount of energy received. A surface whose reflection depends only on the amount of radiance is called Lambertian. In other words, the irradiance does not depend on the viewing angle, and the object looks the same from all camera poses. Lambertian surfaces are assumed in this research [37].

*2.2.2 Projection Theory.* Projection theory describes how light enters into the camera. The *pinhole camera model*, shown in Figure 2.3, is the most simplistic model where only one ray passes into the camera and onto the image plane. A more realistic optic system involves lenses to focus multiple rays, reducing the required exposure time. The *fundamental equation of thin lenses* gives the following relationship for such an optical system:

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f} \tag{2.1}$$

where $Z$ is the distance from the object to the lens, $z$ is the distance from the lens to the virtual image plane, and $f$ is the focal length. Rays of light entering the camera parallel to the lens on one side converge on the focus on the opposite side of the

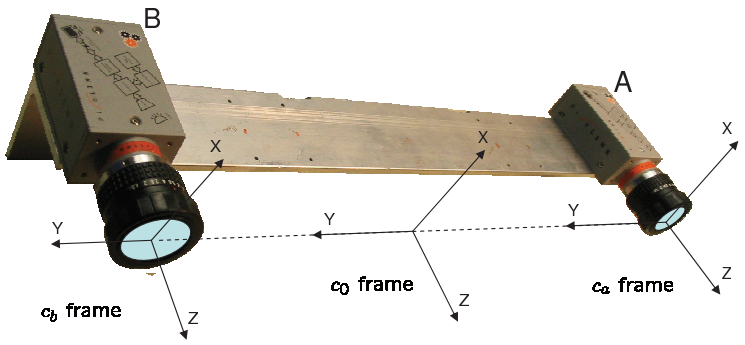



Figure 2.2: Camera Frame Diagram [38]. The camera frames are used to convert from line-of-sight vectors to vectors in the navigational frame. In the binocular case, features are initialized from the neutral point $c_0$. In the monocular case, features are initialized directly from the camera frame, $c_a$.

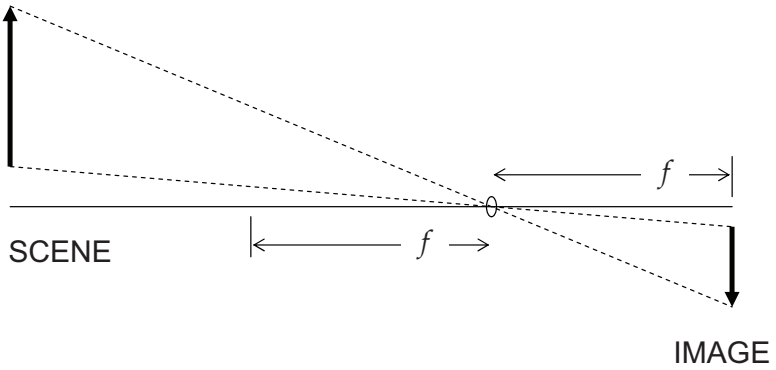



Figure 2.3: Pinhole Camera Model. The pinhole camera model is the most simplistic camera model. Only one ray passes into the camera resulting in a sharp, completely focused image [38].

lens, shown in Figure 2.4. The analysis in this research will use a variation of the pinhole model with the image plane moved to the same side as the scene, as shown in Figure 2.5.

From these basic projection concepts, the intrinsic camera parameters can be defined that transform a spatial scene location into a pixel coordinate ($s^{pix}$). Note that the definition of the camera coordinate system in this research is [down,right]

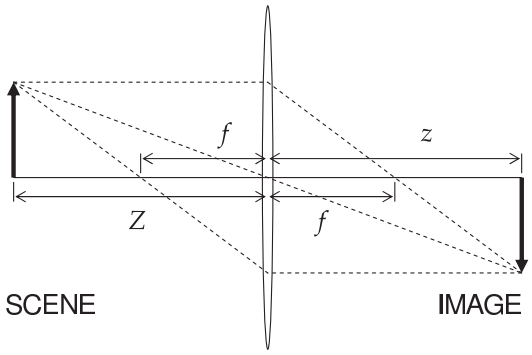



Figure 2.4: Thin Lens Model. The thin lens model is a more realistic projection model. More light enters into the camera reducing exposure times. The fundamental rule for the thin lens model transforms parallel lines into lines passing through the focus on the opposite side of the lens [38].

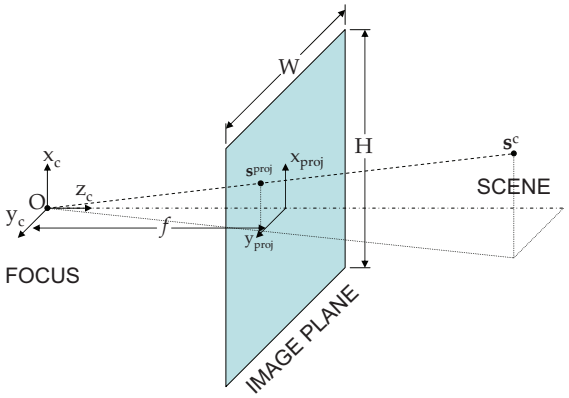



Figure 2.5: Research Camera Model. The camera model assumed in this research is a modification of the pinhole camera model with the image plane on the same side as the scene [38].

instead of the standard computer vision notation, [right,down]:

$$s^{pix} = \begin{bmatrix} -\frac{M}{H} & 0 & 0 \\ 0 & \frac{B}{W} & 0 \end{bmatrix} s^{proj} + \begin{bmatrix} \frac{M+1}{2} \\ \frac{N+1}{2} \end{bmatrix} \tag{2.2}$$

where H and W are the physical height and width of the array, and M and N are the number of horizontal and vertical pixels. $s^{proj}$ is the projection of the scene point onto the image plane and is defined by:

$$s^{proj} = \frac{f}{s_z^c} s^c \tag{2.3}$$

where $s^c$ is the line-of-sight vector in the camera frame, and $s_z^c$ is the z dimension of the line-of-sight vector. Substituting Equation (2.3) into Equation (2.2) and using homogeneous coordinates to incorporate the addition into a matrix multiplication, the result is a single matrix multiply defined by:

$$s^{pix} = \frac{1}{s_z^c} \begin{bmatrix} -f\frac{M}{H} & 0 & \frac{M+1}{2} \\ 0 & f\frac{B}{W} & \frac{N+1}{2} \\ 0 & 0 & 1 \end{bmatrix} s^c \tag{2.4}$$

$$= \frac{1}{s_z^c} T_c^{pix} s^c \tag{2.5}$$

where *intrinsic camera matrix* ($T_c^{pix}$) is given by:

$$T_c^{pix} = \begin{bmatrix} -f\frac{M}{H} & 0 & \frac{M+1}{2} \\ 0 & f\frac{B}{W} & \frac{N+1}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

Figure 2.6 shows the relationship between the projected points and image pixel coordinates.

*2.2.3 Frequency Analysis and Spatial Aliasing.* In this research, information is extracted from the image using an image processing algorithm. If the image is

not properly sampled, the image's information could be corrupted by spatial aliasing causing poor algorithm performance. Fundamentally, the camera is sampling the effectively infinite spectrum of the scene. Spatial aliasing occurs in the camera system when the detectors on the image plane capture frequency content higher than the cutoff frequency defined by the optical system.

Assuming equal distance ($r$) between elements of the CCD, the sampling theorem predicts the maximum observable spatial frequency in the image:

$$f_{max} = \frac{1}{2r} \tag{2.7}$$

The optics act as a low-pass filter when delivering light to the CCD array. The spatial cutoff frequency established depends on the aperture diameter ($D$), light wavelength ($\lambda$), and focal length ($f$):

$$f_c = \frac{D}{\lambda f} \tag{2.8}$$

In typical image setups, the spatial cutoff frequency ($f_c$) is nearly one order of magnitude greater than the maximum sampling frequency defined by the CCD array. This additional frequency content could interfere with the higher captured frequencies. As a result, another low-pass filter is applied in the signal processing software to avoid aliasing effects [37].

*2.2.4 Camera Model and Nonlinearities.* For this research, the camera model refers to the set of parameters that define the characteristics of the camera's intrinsic camera matrix along with a model of the nonlinear deviations from the pinhole camera model caused by the optics. The camera model parameters are estimated using the *Open Computer Vision Toolbox* [5] camera calibration model that considers radial and tangential distortion as the primary nonlinearities. Radial distortion occurs when the camera lens is not the ideal shape causing a distortion around the image perimeter. Tangential distortions occur when the lens is not mounted parallel to the image plane [5] [38]. The method implemented in this package is based on

techniques first introduced in [6]. By removing the nonlinearities, the presented projection theory may be used to transform a point located within the scene into a pixel location, and vice-versa.

With an accurate model of the image acquisition system, image processing algorithms can be applied to retrieve information from the images to be used within the image-aided Kalman filter and stochastic feature tracker.

## *2.3 Fundamental Image Operations*

This section covers basic image operations that are fundamental to the understanding of feature transforms. Features will be the fundamental link between the image and inertial sensors. A feature is a distinct point in the scene comprised of a location and descriptor. Feature transformation is the process that takes an acquired intensity image and produces features.

In this section, the operations required to compute feature transforms are discussed: convolution, noise suppression, gradients, and edge detection. The homographic transform is also introduced in this section. Homographic transformation will be used to aide the matching of feature descriptors in the next chapter.

*2.3.1 Convolution.* Convolution, or in the frequency spectrum multiplication, is a common technique used throughout this research to apply image filters. Since convolution is a linear operation, the associative and distributive properties of the operations can be leveraged to speed up image computations. In this research, Gaussian filter and gradient kernels are applied using convolution. Mathematically, a convolution is defined as:

$$i(i,j) = a * I(i,j) = \sum_{h=-m/2}^{m/2} \sum_{k=-n/2}^{n/2} a(h,k) I(i-h, j-k) \tag{2.9}$$

where $I$ is the image, and $a$ is the kernel matrix applied to a region of (m,n) [37].

*2.3.2 Noise Suppression.* Techniques that use linear and nonlinear filtering to reduce various sources of noise are collectively called noise suppression. The nonlinear median filter is effective against outlier noises, such as shot noise caused by the wave particle nature of light. However, shot noise is not modelled in this research, so median filtering is not implemented.

The most commonly used noise filter is the low-pass filter. The ideal low-pass filter is a two-dimensional sinc function. Two common approximations of the ideal low-pass filter are the averaging and Gaussian filters. The averaging filter effectively cancels some noise by spreading the effects of the noise over the image [37]. A better approximation, the Gaussian filter, has no secondary lobes and is most common approximation used in image processing algorithms [29]. The two-dimensional Gaussian is given by:

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2.10}$$

where $\sigma$ is the standard deviation of the distribution and $x$ and $y$ are the horizontal and vertical dimensions.

In the algorithms described in this research, the Gaussian filter is used as a low-pass filter to reduce high frequency noise. This type of noise is detrimental to derivatives because each differentiation amplifies the noise.

*2.3.3 Gradient/Laplacian.* A gradient ($\nabla$) is a matrix operation that computes the partial derivatives of each dimension. Gradients can also be represented as the magnitude and orientation of the vector of the two partial derivatives. The second partial derivative of each dimension is called the Laplacian ($\triangle$). Orientation information is lost when computing the Laplacian. The gradient and Laplacian are defined mathematically by:

$$\nabla I(x, y) = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \tag{2.11}$$

$$\triangle I(x, y) = \left( \frac{\partial^2 I}{\partial x \partial x}, \frac{\partial^2 I}{\partial y \partial y} \right) \tag{2.12}$$

Gradients are normally implemented by convolving the image with a *kernel* matrix. Two common kernels are the Prewitt and Sobel kernels. The Prewitt kernel for horizontal gradient is given by [**?**]:

$$G_{Prewitt} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \tag{2.13}$$

The Sobel gradient kernel differs from the Prewitt by emphasizing the center pixel. The horizontal Sobel kernel is given by [?]:

$$G_{Sobel} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{2.14}$$

The second dimension's derivative is found by transposing the kernel matrix [25]. The gradient is computed as follows:

$$\nabla I(x,y) = (I_x, I_y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) = \left(G * I(x,y), G^T * I(x,y)\right) \tag{2.15}$$

where $G$ is the Prewitt or Sobel kernel.

*2.3.4 Edge Detection.* Simple edge detection uses the gradient filter's magnitude as the final result. More advanced algorithms, such as the Canny edge detector, use gradient orientation to find edges other than vertical and horizontal. These techniques also use non-maximum suppression to clean up the edges, or dilation and erosion to make continuous edges, called contours [37]. Edge detection will be an important step in the scale-invariant feature transform to improve the algorithms stability during processing. The scale-invariant feature transform computes edges using an eigenvalue decomposition of a local window's Hessian matrix [18].

*2.3.5 Homographic Transform.* This research uses a homographic transform to predict how a image will look after a camera translation and rotation. A homography is a special type of perspective transform between image frames for a planar surface located in the scene. Perspective transforms occur due to the effects of viewing a three dimensional world in a two dimensional image. When rotation and translation occurs together, called a six degree-of-freedom motion (6DoF), the effect perceived by the imaging system is a perspective transformation. In the absence of inertial information, four point correspondences can be used to *estimate* the homography matrix. Each individual point provides two linear constraints on the eight degree of freedom homography matrix. Methods for solving this system are detailed in [13].

In this research, however, the homographic transform ($T_h$) is determined using inertial information. Specifically, the determination uses the interframe rotation ($C_c^{c'}$), translation ($t$) of the camera center, and the intrinsic properties of the camera defined by the intrinsic camera matrix $T_c^{pix}$. Because of the planar assumption, the object can be completely described using the planar normal vector ($n$) and distance from the camera to the plane ($d$). Let $s^c$ be the line-of-sight vector in the camera frame and $s^{pix}$ be the corresponding 2D projection into the image plane, the homographic transform is defined by [35]:

$$s^{c'} = C_c^{c'} s^c + t \tag{2.16}$$

$$T_h = T_{pix}^{c}(C_c^{c'} + \frac{t \otimes n^T}{d})T_c^{pix} \tag{2.17}$$

$$s^{pix'} = T_h s^{pix} \tag{2.18}$$

Figure 2.7 visually demonstrates the homographic transform. Note that the normal vector and planar distance can be appropriately modified for the new frame using the

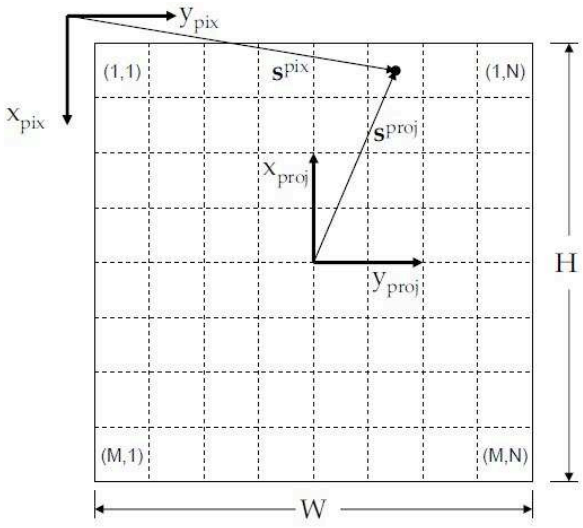



Figure 2.6: Camera Image Array. Camera projection coordinates are transformed into pixels in the image array of (MxN) pixels. H and W represent the physical height and width of the array [38].

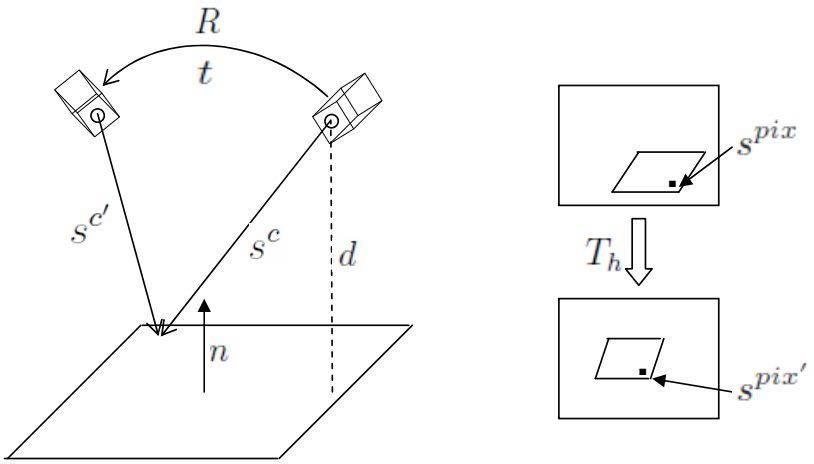



Figure 2.7: Homographic Transform. The homographic transform is computed using the camera center rotation ($C_c^{c'}$) and translation ($t$) along with the planar normal vector ($n$) and the distance from the camera center to the plane ($d$). For every point on the plane, the transformation between frames is defined by the homography ($T_h$).

following relation [35]:

$$n' = C_c^{c'} n \tag{2.19}$$

$$d' = d - n^T t \tag{2.20}$$

## *2.4 Feature Transforms*

Features are fundamental to how this research fuses imaging and inertial sensors. Distinct locations in the scene are tracked to determine the relative position of the aircraft. Feature transformation is the process of detecting these *interest points* and extracting a description from a local image region around that interest point. Feature detection normally uses image derivatives described in Section 2.3.3. Feature extraction can consist of capturing intensity or gradient information. In the case of the scale-invariant feature transform (SIFT) however, additional computation is required to determine the descriptor.

Three feature detection algorithms are reviewed in this section: Harris, Good Features, and SIFT. The section section covering SIFT also discusses feature extraction.

*2.4.1 Harris Corner Detector.* First formalized in 1988 by Harris and Stephens, the algorithm computes an image gradient matrix ($I_g$) whose elements are smoothed by a Gaussian kernel ($g(\cdot)$, see Equation (2.10)) for a local window. First, the derivative image is computed for each dimension:

$$I_x(x, y) = G * I(x, y) \tag{2.21}$$

$$I_y(x, y) = G' * I(x, y) \tag{2.22}$$

Next, the gradient matrix and its eigenvalues ($\alpha$,$\beta$) are computed:

$$I_g(x,y) = \begin{bmatrix} \sum g * I_x^2 & \sum g * I_x I_y \\ \sum g * I_y I_x & \sum g * I_y^2 \end{bmatrix} \quad \Rightarrow \quad eig(I_g) = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \tag{2.23}$$

The size of the eigenvalues ($\alpha$,$\beta$) of $I_g$ determines the nature of the surface. Two large eigenvalues equate to a strong corner, with significant gradient in both directions. Weaker corners have one or two small eigenvalues. A single large eigenvalue indicates information in only one direction and implies an edge. This condition is known as the *aperture problem*, where only the distance perpendicular to the edge is detectable [37]. Two small eigenvalues correspond to little information, or a constant intensity surface. In either case, the feature location cannot be uniquely determined.

Instead of computing a full eigenvalue decomposition and comparing eigenvalues directly, the following quality metric is used [12]:

$$C(x,y) = \det(I_g) - k(\text{trace}(I_g))^2 \tag{2.24}$$

where:

$$\det(I_g) = \alpha\beta \tag{2.25}$$

$$\text{trace}(I_g) = \alpha + \beta \tag{2.26}$$

This scalar metric is thresholded to determine corners, and the tuning parameter $k$ can be varied from 0 to 1. Smaller values of the tuning parameter favor two large eigenvalues to produce a high metric score. The parameter $k$ is commonly set to 0.4, determined empirically [8]. Corners, the feature detection output, are points in the quality metric matrix that exceed a constant threshold [12].

The Harris corner detector is invariant to rotation but not scaling changes [33]. Improvements to the Harris corner detector include eliminating the tuning parameter

$k$ by using a ratio of image gradients [26] and adding a scale-space search [31] similar to the later discussion of the scale-invariant feature transform.

*2.4.2 Shi Tomasi Good Features Detector.* In [34], Shi and Tomasi state that the performance of a feature detection algorithm is tied closely to the type of tracking algorithm used. For their application, the Kanade-Lucas tracking algorithm [19], a so-called "*Good Feature*" involves checking invertibility for a least-squares tracking solution. This does not explicitly guarantee a corner in the strict sense. Instead, the algorithm computes the minimum eigenvector of the local gradient window around each pixel. The threshold for acceptable features is a percentage of the *global maximum* of the set of each pixel's minimum eigenvalue. Points higher than this dynamic threshold are candidate features. Finally, a minimum feature distance between local maximums determines which features are kept. The Good Features detection algorithm is summarized by the following steps [4]:

1. Determine the minimum eigenvalue for each pixel's local gradient window $I_g$
2. Determine the *maximum* of all minimum eigenvalues
3. Identify features above a percentage of the *global maximum* minimum eigenvalue (i.e., the feature quality)
4. Determine local maximums from the remaining features within a predefined minimum distance of other local maxima

The Harris corner and Good Features detectors are generally referred to as low-level feature detection. Next, the more complex scale-invariant feature transform is introduced.

*2.4.3 Scale-Invariant Feature Transform .* The scale-invariant feature transform (SIFT) is considered a modern feature detection algorithm that is invariant to scale, rotation, and partially invariant to changes in illumination and affine warping. Because of these characteristics, SIFT has found its way into many applications

including pattern recognition, structure from motion, stereo correspondence, and motion estimation.

In this explanation of SIFT, *keypoints* are synonymous with interest points defined previously. The algorithm is comprised of four stages:

1. Scale-space extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor computation

In this research's context, the first two stages are considered feature detection, and the final two stages are feature extraction. Scale-space extrema detection uses a difference of Gaussian (DoG) filter computed at a fixed number of scales per octave. The DoG is an approximation of the scaled-normalized Laplacian of Gaussian, whose local maxima and minima are stable features [18]. The DoG is given by:

$$D(x, y, \sigma) = [g(x, y, k\sigma) - g(x, y, \sigma)] * I(x, y) \approx (k - 1)\sigma^2 \nabla^2 g(x, y, \sigma) \qquad (2.27)$$

where $x, y$ are the two dimensional image location, $\sigma$ is the standard deviation, $g$ is the Gaussian kernel, $k$ is the scale parameter, and $I$ is the image. At each octave, the image is downsampled by a factor of two, effectively doubling the standard deviation. Determined empirically, three scales per octave was found to produce adequate scale sampling. Experiments found that further sampling would lead to more extrema but increasing instability, along with increased cost of computation (more convolution). Experiments also showed an appropriate choice for the Gaussian smoothing to be a standard deviation of 1.6. This was chosen as a tradeoff between feature repeatability and speed (size of the convolution window). The smoothing effectively discards the highest frequency content (see Section 2.2.3), but the algorithm avoids information loss by upsampling the image prior to smoothing. A three-dimensional search space

is now available to localize extrema and determine keypoints. Figure 2.8 shows an example scale-space decomposition.

The determination of keypoints distinct from their neighbors (i.e., the search for high contrast) involves a gradient strength comparison of the nine elements on either side of the keypoint's scale and the eight neighbors in the same scale. Once an extrema is found, interpolation in the three dimensional search space is used to improve the location and scale estimates. Edge responses are also discarded to increase the stability of detected features. According to the author, features found along edges are sensitive to noise because of the second derivative technique of localization.

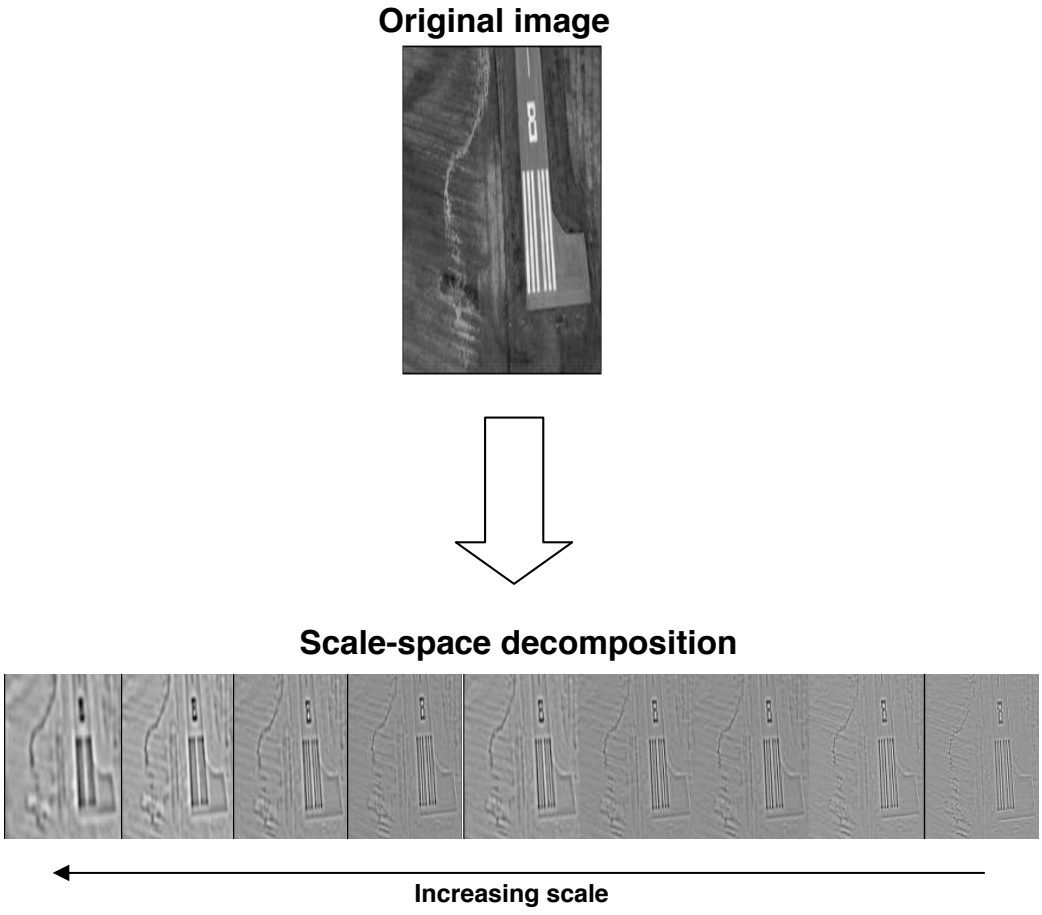



Figure 2.8: Scale-space Decomposition. A captured image in decomposed into multiple scales using the difference of Gaussian filter. Local extrema are detected in this three dimensional search space. Images from [38].

Orientation assignment occurs in a scale-invariant method, using the filtered image at the keypoint's scale for calculations. Sample points of gradient orientation are weighted with a Gaussian filter centered around the keypoint then fit into a histogram with 10 degree bins. The primary orientation is assigned according to the three closest histogram values to each peak [18].

With location, scale, and orientation determined, the descriptor computation attempts to reduce the effects of viewpoint and illumination change. Inspired by biological research into the visual cortex, the method incorporates gradient and spatial frequency. The descriptor coordinates are determined relative to the primary orientation. This relative definition of coordinates achieves rotation invariance in the descriptor. The local gradient orientations are apportioned to 4x4 subregion histograms, each with 45 degree bins. The values from each histogram are placed into a 128 element vector (4x4x8). Normalization is applied to the vector to provide illumination invariance.

The final result of the processing is location, scale, primary orientation, and histogram vector. The feature *descriptor* is the scale, primary orientation, and histogram vector [17] [18].

Significant processing is required to determine SIFT's robust features. There have been a number of attempts to increase the speed of the algorithm using techniques such as principle component analysis (PCA) [15]. Alternatively, SIFT inspired algorithms of comparable performance also exist, e.g. SURF [1] [2].

### *2.5 Feature Matching*

After features are detected and extracted, the next step in feature tracking is the matching of descriptors. The feature matching techniques reviewed in this section includes Euclidean distance, normalized cross correlation, and gradient techniques. SIFT uses Euclidean distance between one-dimensional feature descriptors to match features. During low-level feature detection algorithm discussion, a feature descriptor

was not specified. Commonly, the two-dimensional local intensity or gradient is used as the local descriptor. The descriptor selected for this research is the local intensity and is matched using normalized cross-correlation. Gradient techniques are also discussed for possible future improvements and testing.

*2.5.1 Euclidean Distance.* Previous research [38] used a 1x128 element descriptor from SIFT. Matching two SIFT descriptors involves computing a Euclidean distance. Given two descriptors $a$ and $b$, the Euclidean distance ($d_E$) is given by:

$$d_E = \| a - b \|_2 \tag{2.28}$$

where $\| \cdot \|_2$ is the two-norm. Using a small angle approximation [18], calculation is simplified:

$$d_E \approx \arccos < a, b > \tag{2.29}$$

Since the feature descriptors are normalized and non-negative, the result should range from 0 to $\frac{\pi}{2}$, with smaller Euclidean distances corresponding to stronger matches.

Next, two-dimensional matrix techniques for low-level feature extraction are introduced, beginning with the normalized cross-correlation of intensity windows.

*2.5.2 Normalized Cross-Correlation.* Cross-correlation is used in matching because of its similarity to the Euclidian distance, however normalization is necessary to properly match templates in the presence of illumination changes and saturation. The normalized version of the cross-correlation is commonly referred to as the *correlation coefficient* or *normalized cross-correlation* (NCC). The correlation coefficient ($\rho$) between the template ($t$) and the comparison window ($w$) for a location $(u, v)$ within

the larger image is given by:

$$\rho(u,v) = \frac{\sum_{x,y}[w(x,y) - \bar{w}_{u,v}][t(x-u,y-v) - \bar{t}]}{\sqrt{\sum_{x,y}[w(x,y) - \bar{w}_{u,v}]^2 \sum_{x,y}[t(x-u,y-v) - \bar{t}]^2}} \tag{2.30}$$

In [16], a more efficient computational method is introduced. The correlation score ranges from -1 to 1. A Gaussian filter may be applied over the image to de-emphasize the regions of the correlation most affected by misregistration [18]. This primarily deals with boundary regions of the image.

*2.5.3 Gradient Techniques.* Gradients are calculated in search of features and are nicely invariant to uniform intensity shifts. When normalized, the gradient is also invariant to intensity scaling. These normalized gradient matching techniques, such as [41], claim increased accuracy in the presence of strong illumination changes.

Similar to the building of the SIFT descriptor (see Section 2.4.3), research in [11] proposes that gradient orientation be used in correspondence matching. The authors show the technique is computationally inexpensive for small templates, with improved correlation results. However, the techniques still do not provide rotation or affine invariance for the descriptor.

## *2.6 Motion Estimation*

In this research, estimation of aircraft pose and feature locations occur within the Kalman filter, an optimal estimator. In this section, vision-only techniques of estimation are introduced and contrasted with Kalman filtering.

*2.6.1 Least-Squares Estimation.* In most vision-only applications, the fundamental matrix ($F$), or essential matrix ($E$) if the camera calibration ($T_c^{pix}$) is known,

is estimated to determine camera motion from one frame to another:

$$x'Fx = 0 \tag{2.31}$$

$$E = T^{c}_{pix} F T^{pix}_{c} \tag{2.32}$$

where $x$ is the set of pixel locations in the first frame and $x'$ is the pixel locations in the second frame. This relationship is a constraint of projective geometry and states that the cross product of a point with itself (when transformed back into the same frame) is zero. The fundamental matrix has seven degrees-of-freedom and is usually determined using a least-squares method. Problems can plague this method of determining motion while using noisy measurements. First, the estimated solution does not have to correspond to any real-world motion. Real-world motion can be defined as a rotation, translation, and perspective transform [14]. Additionally, this method is susceptible to the incorporation of erroneous measurements [13].

In this research, inertial sensor measurements eliminate the need to determine the camera motion by vision-only estimation. Instead, inertial and image measurements are optimally combined using a Kalman filter.

*2.6.2 Kalman Filtering.* Kalman filtering seeks to determine the solution of stochastic differential equations modelling system dynamics while incorporating discrete measurements. Linear and nonlinear Kalman filters are introduced in this section. For information beyond this discussion on Kalman filtering see [7] or [21] [22].

*2.6.2.1 Linear Kalman Filtering.* Linear Kalman filtering optimally solves the linear stochastic differential equation of the form:

$$\dot{x} = Fx + Bu + Gw \tag{2.33}$$

with the systems dynamics matrix $F$, state vector $x$, input matrix $B$, input vector $u$, the noise matrix $G$, and the noise sources $w$. The noise sources are all assumed

zero-mean, white Gaussian noise processes. The covariance of the noise is defined by:

$$E\{w(t)w^T(t+\tau)\} = Q(t)\delta(\tau) \tag{2.34}$$

where $E\{\cdot\}$ is the expectation operator, $Q$ is the process noise, and $\delta$ is the dirac delta function. In addition, each state is a Gaussian noise process, whose statistical distribution is completely defined by a mean ($x$) and covariance ($P$). The time propagation of these statistics are defined by the following equations:

$$x(t_i^-) = \Phi(t_i, t_{i-1})x(t_{i-1}^+) + \int_{t_{i-1}^+}^{t_i^-} \Phi(t_i, \tau)B(\tau)d\tau \tag{2.35}$$

$$P(t_i^-) = \Phi(t_i, t_{i-1})P(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + \int_{t_{i-1}^+}^{t_i^-} \Phi(t_i, \tau)G(\tau)QG^T(\tau)\Phi^T(t_i, \tau)d\tau \tag{2.36}$$

where $t_i^-$ is the instant in time just before the increment of time ($i$), and $t_i^+$ is the instant immediately after. $\Phi$ is the state transition matrix determined by the matrix exponential:

$$\Phi(t_i, t_{i-1}) = e^{F(t_i - t_{i-1})} \tag{2.37}$$

Discrete state measurements are incorporated using a measurement model and Kalman filter update equations. The measurement model is defined by:

$$z(t_i) = H(t_i)x(t_i) + v(t_i) \tag{2.38}$$

$$E\{v(t_i)v(t_j)^T\} = R(t_i)\delta_{ij} \tag{2.39}$$

where $z$ is the discrete measurement, $H$ is the influence matrix, $v$ is zero-mean Gaussian noise, $R$ is referred to as the measurement noise, and $d_{ij}$ is the Kroeneker delta function. A filter update involves first computing the Kalman filter gain $K$, a measure

of the certainty of the measurement over the propagated estimate:

$$K(t_i) = P(t_i^-)H^T(t_i)[H(t_i)P(t_i^-)H^T(t_i) + R(t_i)]^{-1} \tag{2.40}$$

The measurement is then incorporated into the filter estimate using the update equations:

$$x(t_i^+) = x(t_i^-) + K(t_i)\left[z(t_i) - H(t_i)x(t_i^-)\right] \tag{2.41}$$

$$P(t_i^+) = P(t_i^-) - K(t_i)H(t_i)P(t_i^-) \tag{2.42}$$

Together the update and propagation define the optimal solution to the stochastic differential equation for all time, given the filter assumptions of linearity and noise properties are not violated [21] [38].

*2.6.2.2 Extended Kalman Filter.* Nonlinear models violate the assumptions of the conventional linear Kalman filter. In the extended Kalman filter (EKF), the nonlinear model is linearized around the current nominal state estimate. This section introduces the basics of extended Kalman filtering.

Consider stochastic differential and measurement equations:

$$\dot{x}(t) = f[x(t), u(t), t] + Bu(t) + Gw(t) \tag{2.43}$$

$$z(t_i) = h[x(t_i), t_i] + v(t_i) \tag{2.44}$$

where $f$ and $h$ are nonlinear functions of the state $x$ and input $u$. The goal of the extended Kalman filter is to develop linear state propagation and measurement equation as shown in the conventional Kalman filter. The nonlinearity is approximated using a Taylor series expansion and perturbation model. The result of this analysis is a whole-valued state composed of the current optimal estimate $(\bar{x}(t), \bar{z}(t))$ and a

perturbation $(\delta x(t), \delta z(t))$.

$$x(t) = \bar{x}(t) + \delta x(t) \tag{2.45}$$

$$z(t) = \bar{z}(t) + \delta z(t) \tag{2.46}$$

The state perturbation is propagated in time according to the following equations:

$$\delta \dot{x}(t) = F(t)\delta x(t) + Gw(t) \tag{2.47}$$

$$F(t) = \frac{\partial f[x(t), u(t), t]}{\partial x}\Big|_{x=x_n(t), u=u_n(t)} \tag{2.48}$$

where $F$ is the linearization of the dynamics around the current optimal state estimate and inputs, or nominals. Similarly, the measurement equation is linearized:

$$\delta z(t_i) = H(t_i)\delta x(t_i) + v(t_i) \tag{2.49}$$

$$H(t_i) = \frac{\delta h[x(t_i), t_i]}{\delta x}\Big|_{x=x_n(t_i)} \tag{2.50}$$

where $H$ is the linearized measurement equation around the nominal estimate. Now the conventional Kalman filter propagation and update equations can be used to estimate the perturbation in time, and the whole valued states can be determined using Equations (2.45) and (2.46). This filter derivation provides a biased estimate, and the solution is no longer completely optimal [21].

In the next section, related research is presented and discussed including the previous research at the Air Force Institute of Technology.

## *2.7 Related Research*

This section covers similar techniques that use inertial and camera measurements to improve small vehicle navigation. First, an overview of the previous research at AFIT is presented. Similar findings are also presented in the field of image and in-

ertial sensor fusion. Finally, techniques that served as the motivation for the low-level feature descriptor aiding are introduced.

*2.7.1 Overview of Previous Research.* Previous research at the Air Force Institute of Technology demonstrated the fusion of image and inertial sensors using a technique called stochastic feature tracking [38]. Features are distinct points within the scene comprised of a location and descriptor. An extended Kalman filter tracks aircraft position, velocity, and attitude along with the location of stationary features, called *landmarks*. The current image frame's features are matched to landmarks using a correspondence search constrained by each landmark's current uncertainty. The reduction of the search space using uncertainty is termed the stochastic constraint. For the previous research, the scale-invariant feature transform (SIFT) [18] was selected for its robustness over diverse camera movements. This research attempts to find a less computationally intensive feature transform aided to be more robust and still stochastically constraining the correspondence search. Further details of the algorithm are reviewed in Section 2.8 and Chapter 3.

*2.7.2 Image and Inertial Sensor Fusion.* A number of similar feature tracking systems have been attempted for use in the flight control of an aircraft. In [20], an extended Kalman filter with image, inertial, and magnetometer measurements was used as a navigation reference for a simulated helicopter. The simulated environment used basic shapes to simplify feature extraction and assumed that the feature's location was known. The experiment verified that images were able to constrain inertial drift and successfully navigate the helicopter.

In [9], the author used a fisheye lens to capture images and extract features by using a rotationally-fixed projection with a Harris corner detector [12]. This type of extraction allowed for features to be tracked over an extended period of time in simulated and real imagery. Over a closed-loop trajectory, the results showed a reduction overall position error when using feature tracking. This research will use real imagery

and more conventional camera setups attempting to track features using alternative techniques described in the next section.

*2.7.3 Deeply-Integrated Imaging and Inertial Sensors.* The deep integration of imaging sensors involves prediction of feature descriptors using inertial information. In [3], feature tracking with a conventional camera is improved using an inertial sensor. Feature descriptors are *derotated*, or rotationally warped, between captured frames using inertial information to improve feature matching.

Techniques in [35] involve using inertial and imaging sensors to stabilize an aircraft in hover. Simulations verify the control law development based on the concept of a image homography. A homography can predict how a planar surface will look from a different camera pose, and this paper introduced a homography derivation using inertial information. In this research, this homography formulation will be used to transform feature descriptors into a new camera pose for matching.

In the next section, the image-aided Kalman filter developed in previous research is covered in more detail.

## *2.8 Image and Inertial Fusion Algorithm*

In this section, the previous image-aided Kalman filter (IAKF) is described in detail. The previous research developed an EKF with three measurement updates: alignment, inertial, and image. Stationary alignment updates are used to allow the filter to estimate biases in the accelerometers and gyros. The inertial update reads the rates from the gyros and accelerations from the accelerometers, and this serves as the core update for many Kalman filters. This section reviews concepts specific to the image update. During the image update, tracked positions of stationary features, called *landmarks*, are estimated within the filter. This section reviews how relative position in the camera frame is converted into the navigation frame and the general components of the stochastic image update, also known as the stochastic feature tracker.

*2.8.1 Determination of Landmark Location.* During landmark initialization, a line-of-sight vector from the camera frame is determined ($s^c$). This vector is converted into the navigation frame using the camera-to-body DCM ($C_c^b$), body-to-navigation DCM ($C_b^n$), and the current camera position $p^n$:

$$s^n = C_b^n C_c^b s^c \tag{2.51}$$

$$t^n = p^n + s^n \tag{2.52}$$

The result is the landmark location in the navigation frame ($t^n$). Figure 2.9 demonstrates this process visually.

*2.8.2 Landmark Uncertainty Initialization.* An important contribution of the previous research [38] was the development of the landmark's uncertainty based on the statistics of the information used to derive the location. In general, this involves computing partial derivatives with respect to each state estimate used in the determination of the landmark's initial location estimate. In addition, this aggregation assumes that these error sources are independent. The section will review two spe-

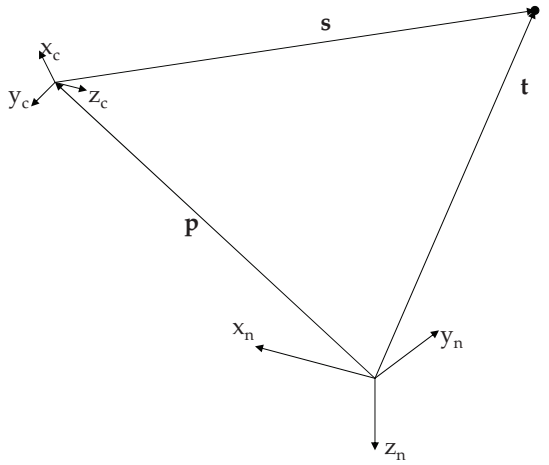



Figure 2.9: Determination of Landmark Location [38]. The camera position line-of-sight ($s$) vector is converted into a target location ($t$) using the current camera position ($p$), and camera to navigation frame DCM ($C_c^n$).

cific methods of determining a landmarks location based on monocular and binocular initialization.

In monocular vision, additional information is necessary to determine a landmark's location due to the unobservable scale parameter. Determination of this scale parameter in this research uses a method of egomotion with an initial guess at the depth of features. A large uncertainty will allow feature depth to be estimated accurately after some motion and matching of the landmark. The chosen depth acts as a gain on the amount of motion, with a larger depth allowing subtle motion. The uncertainty defines the search space around the predicted location.

Specifically the landmark's location is determined by:

$$t^n = p^n + \underbrace{C_b^n \left[ d C_c^b T_{pix}^c \underline{z} \right]}_{s^n} \tag{2.53}$$

where $p^n$ is the current aircraft location, $C_b^n$ is the current aircraft attitude DCM, $d$ is the mean depth of features in the scene, $C_c^b$ is the camera to body DCM, $T_{pix}^c$ is the intrinsic camera matrix, and $\underline{z}$ is the homogenous pixel location. The initial landmark position $t^n$ uncertainty is a composite of the uncertainties in the parameters of the measurement equation. Assuming independence of the parameters, the composite uncertainty is computed by the following equation:

$$P_{tt} = G_{tp} P_{pp} G_{tp}^T + G_{t\psi} P_{\psi\psi} G_{t\psi}^T + G_{td} P_{dd} G_{td}^T + G_{t\alpha} P_{\alpha\alpha} G_{t\alpha}^T + G_{tz} P_{zz} G_{tz}^T \tag{2.54}$$

where $\psi$ and $\alpha$ correspond to the attitude error vectors of the $C_b^n$ and $C_c^b$ matricies. The influence matricies ($G$) determine how much each component uncertainty factors into initial landmark uncertainty. These influence matricies are computed by taking partial derivatives with respect to each parameter. For a scalar depth parameter ($d$),

the influence matricies are specified by:

$$G_{tp} = \frac{\partial t^n}{\partial p^n} = I_{3x3} \tag{2.55}$$

$$G_{t\psi} = \frac{\partial t^n}{\partial \psi} = C_b^n \text{ skew}(\left[dC_c^b T_{pix}^c z\right]) \tag{2.56}$$

$$G_{dd} = \sigma_d^2 \tag{2.57}$$

$$G_{t\alpha} = \frac{\partial t^n}{\partial \alpha} = -dC_b^n C_c^b \text{ skew}(\left[T_{pix}^c z\right]) \tag{2.58}$$

$$G_{tz} = \frac{\partial t^n}{\partial z} = dC_b^n C_c^b T_{pix}^c \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{2.59}$$

where $\sigma_d$ is the standard deviation of the static feature depth. This is a simplification from the work presented in previous research because the terrain reference is not a function of the slant angle.

Binocular initialization involves no *a priori* information about the environment and determines the location of a landmark using the disparity between the cameras. Landmarks are determined according to the following equation:

$$y^n = p^n + C_b^n \left[p_0^b + C_{c0}^b s_0^{c0}\right] \tag{2.60}$$

where $y^n$ is the landmark location for a binocular initialized feature and $s_0^{c0}$ is the line-of-sight vector from a neutral frame between the two cameras. Figure 2.10 illustrates this feature initialization geometry. During this initialization, a candidate feature from the first camera is matched to a feature in the second camera using a stochastic search space defined by the binocular disparity. Once a match is determined, a linear regression is used to determine the neutral line-of-sight vector ($s_0^{c0}$) from the line-of-sight vector in each camera. Once determined, the result is substituted in to Equation (2.60) to determine the estimated location in the navigation frame $y_n$. Determination of the uncertainty in the measurement resembles computation of partial derivatives in the monocular case. See [38] for the full derivation.

*2.8.3 Stochastic Constraint.* Figure 2.11 illustrates the process of matching landmarks with the stochastic constraint. The image update begins by detecting and extracting features in the current frame. Next, each landmark and its uncertainty is propagated to the current time. The predicted location and uncertainty are projected into the camera frame, and features falling within this region are matched to landmarks. The stochastic constraint refers to the two sigma uncertainty search space based on the current landmark. A feature is matched if the matching metric is higher than a specified threshold. From a matched landmark's location, a residual is computed that is incorporated into a Kalman filter update. Finally, the process of landmark administration refers to the process of identifying features that have not been matched recently. These *stale* landmarks are replaced by new features selected by computing feature quality metric based on the Mahalanobis distance from other currently tracked landmarks.

This concludes the background material required to develop the deeply-integrated feature tracking algorithm. In the next chapter, the development methodology will be introduced including the selection of the low-level feature transform and the necessary inertial aiding of the low-level descriptor.

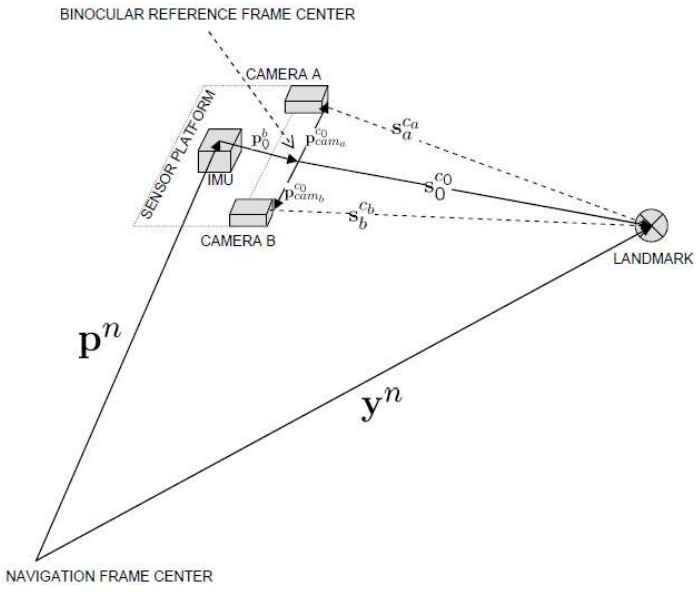



Figure 2.10: Binocular Feature Initialization [38]. During binocular initialization, features are initialized from a neutral point between the cameras ($c_0$). This feature initialization requires no *a priori* information.

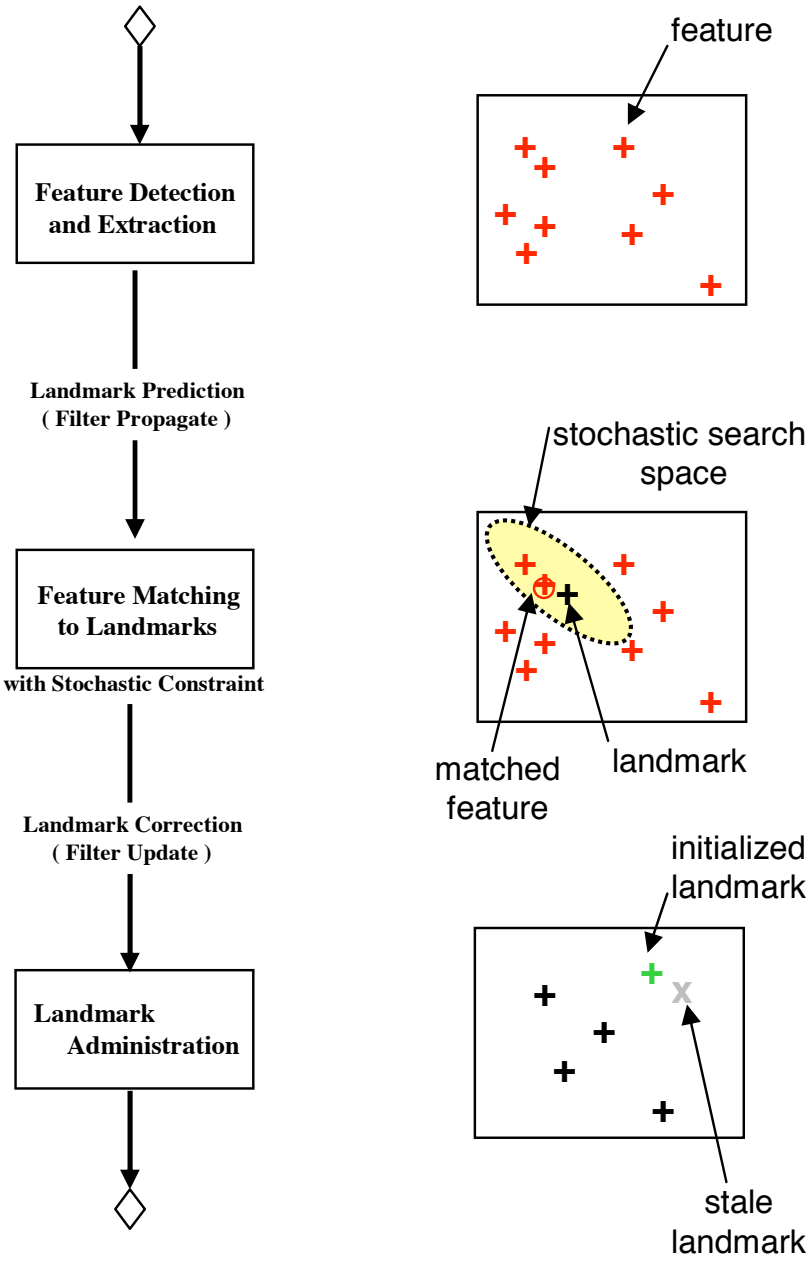



Figure 2.11: Extended Kalman Filter Image Update. The image-aided Kalman filter's stochastic feature tracking, or image update, involves three steps: feature detection/extraction, landmark matching (with the stochastic constraint), and landmark administration.

# III. Methodology

This section introduces the methodology used to develop a deeply-integrated feature tracker for embedded navigation. First, the feature trade space is introduced, and a new low-level feature extraction algorithm is selected. Next, the modifications to the stochastic feature tracker within the context of the image-aided Kalman filter are discussed.

The next section introduces deep-integration methods used to aid the low-level feature descriptor, including rotation and six degree-of-freedom motion (6DoF) motion aiding. Finally, a section addressing monocular feature location initialization is discussed. Prior to this discussion, binocular initialization is assumed. Monocular initialization was selected for the final implementation for its weight and computation savings for a indoor aircraft hover experiment.

## *3.1 New Feature Transformation Selection*

*3.1.1 Feature Trade Space.* The first two steps of feature tracking are the feature detection and extraction, together called a *feature transform*. The result of a feature transform is a feature location and local description, called a *descriptor*. In the absence of additional sensor information, the most desirable feature transformation algorithm would completely separate a feature's location from its descriptor. No known computer algorithm achieves this invariance entirely, but the scale-invariant feature transform (SIFT, see Section 2.4.3) achieves a high degree of invariance. A human's visual processing capability closely approaches the ideal. Consider the following example. Given a pen or pencil found on your desk, blindly move the object across your desk as in Figure 3.1. Quickly you will be able to find the where you moved the object. In this case, the object is a red pen, and you recognized the pen despite its location in the environment. This occurs because the description of the object does not depend on its location. In fact, human visual perception is invariant to a number of different image warping. Figure 3.2 shows warping examples that

may occur when you move the pen. In any of these case, humans can distinguish the identical object or feature where image processing may not.

This variety of feature location and descriptor invariance is shown as a spectrum of feature transformation algorithms in Figure 3.3. This research proposes that with inertial sensor information, this lack of invariance in low-level feature descriptor can be compensated to achieve results of a feature transformation that is nearly invariant.

*3.1.2 Feature Transform Selection.* Next, the feature transformation problem is decomposed into two steps: feature detection and extraction. Feature detection determines the location of the interest point in the environment. Feature extraction computes the feature's description.

Two traits are desirable for the feature detection algorithm: strength and repeatability [33]. Finding strong or dominant features is important to avoid background clutter and separation from other features. Repeatability depends on the strength of a feature when viewed from different camera poses. Common feature detection algorithms use gradients to determine the interest points in the image. This is

(a) (b)

Figure 3.1: Human Visual Processing Example. A red pen place on a desk in the image on the left. The pen is randomly moved and rotated in the image on the right. Humans can identify this object easily because of an invariance of the description to the location.

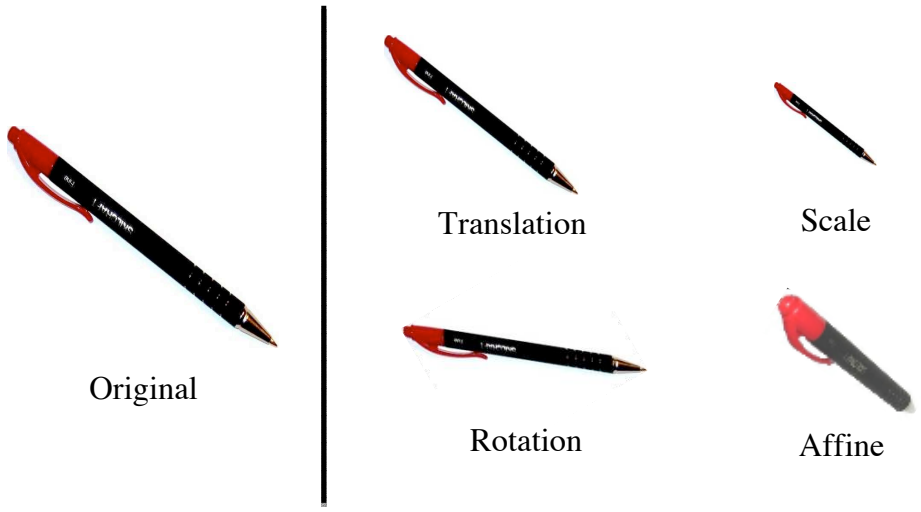



Figure 3.2: Image Warping Examples. Each warping effect of a pen on the right can be identified as the original pen on the left due to a human's ability to keep the object's location and description independent.

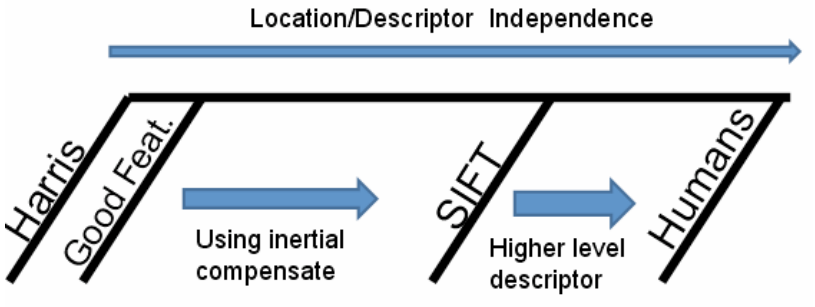



Figure 3.3: The Feature Spectrum. In low-level feature description algorithms, relationship to the feature's location significantly affects the description. SIFT does considerable computation to reduce the effect of scale and rotation on its descriptor. In the ideal relationship, a feature's location is entirely invariant of its location. No known feature transform is capable of achieving this level of invariance, but humans are good example of a high level of invariance.

the case with the three feature transformation algorithms considered in this research: the Harris corner detector, Good Features detector, and the scale-invariant feature transform. These algorithms, however, differ in their localization of gradient maximum. The first two algorithms use gradient magnitude in the image plane, while SIFT incorporates scale into the search space. These algorithms detect features by comparing peaks to other nearby peaks as well as a threshold value. Repeatability in feature detection is closely tied to the strength of the corner over different viewing angles.

The Good Features and Harris algorithms are similar in their low-level detection of interest points. Good Features detection was eventually selected for this research because it produced a repeatable and distinct group of features over the experimental environments. The dependable detection of features can be attributed to two important differences in the algorithms. The global thresholding in the Good Features algorithm allows real-time adjustment of the feature detection threshold that avoids feature starvation conditions (see Section 2.4.2). Feature starvation is the inability to add new feature to track causing the navigational estimate to drift. Secondly, Good Features are more repeatable over 6DoF camera motions [34].

Besides feature detection, the low-level feature algorithms distinguish themselves from SIFT also in their descriptor, formed through feature extraction. Although many local descriptors could be associated with a low-level feature detection, this research takes the lowest-level description, local intensity. There is no calculation for this feature descriptor. For SIFT, dominant gradient orientation is used to form the descriptor, also accounting for scale and rotation. This requires significantly more computation.

*3.1.3 Tuning the Good Features Detection.* The Good Features algorithm allows for some adjustment of feature quality, spacing, and amount of detections (see Section 2.4.2). For every pixel in the image, the minimum eigenvalue of a local image gradient is computed. Each minimum eigenvalue is compared to a percentage of the

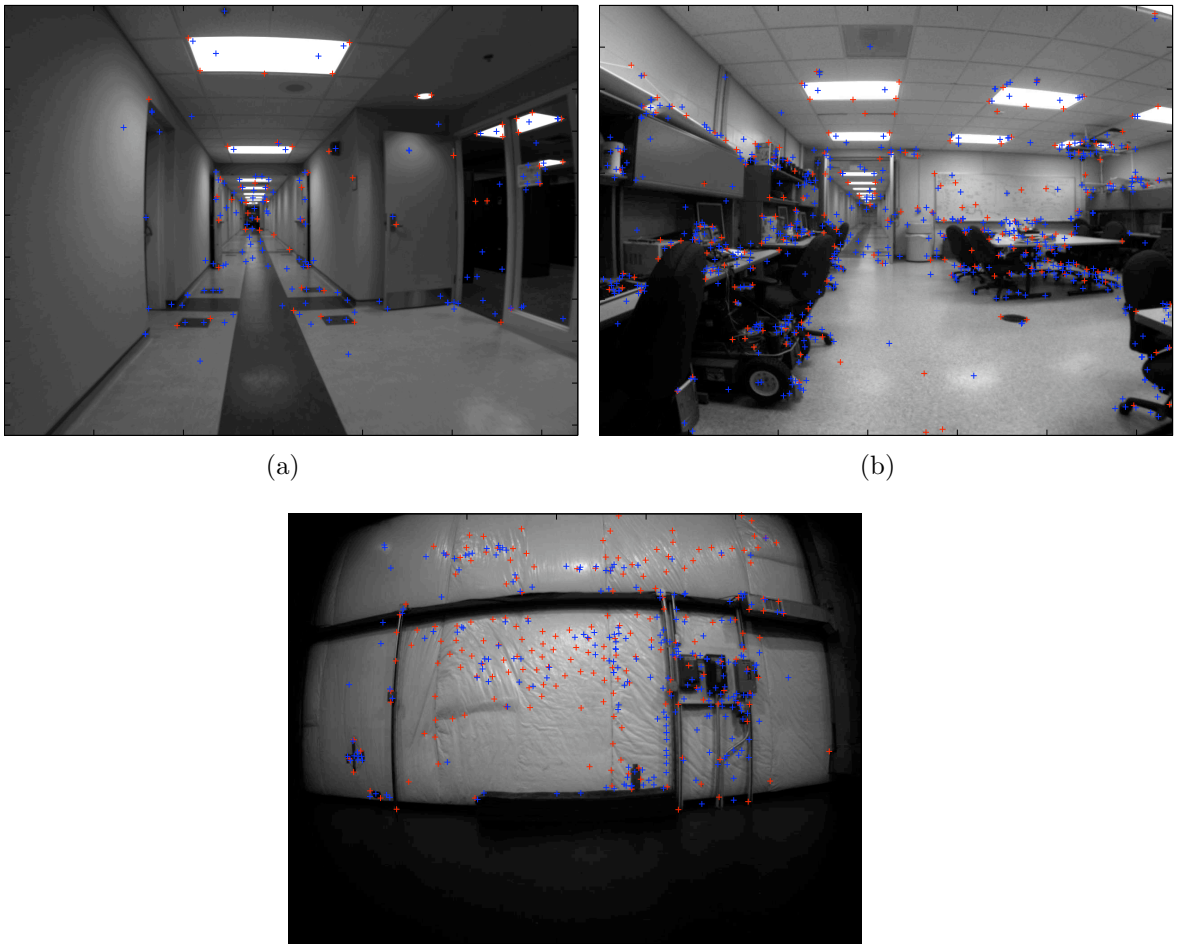
max of these minimum eigenvalues. The percentage is the Good Features quality. Each candidate feature is compared to others within the spacing radius, and only the strongest candidate feature is kept. Finally, only a specified number of features are kept to avoid lengthly processing times. A quality of 0.01, spacing of 10 pixels, and a maximum of 200 features detected a desirable set of features. These numbers were empirically determined through multiple runs in the simulated filter. Selecting a higher quality metric would result in increased feature strength, but a decrease in repeatability. The feature spacing also helped with repeatability of strong features over different camera motions.

In Figure 3.4, SIFT and Good Feature detections are compared on a set of images. Notice that SIFT specifically chooses features that are not located along edges, while Good Features simply takes the strongest magnitude features in the image.

## *3.2 Feature Tracking with Good Features*

This research centers around a replacement stochastic feature tracker for the previously developed image-aided Kalman filter. Figure 3.5 shows the feature tracker in relation to the extended Kalman filter propagation and inertial mechanization.

Features tracked in the filter are referred to as *landmarks*. Landmarks are tracked and initialized in accordance with the image fusion algorithm (see Section 2.8). The state vector contains fifteen elements, plus additional states for landmark position estimation. The first fifteen states are defined as the navigation state error vector ($\delta x$) comprised of position ($\delta p^n$), velocity ($\delta v^n$), attitude ($\psi$), accelerometer bias ($\delta a^b$), and



(a) (b)

(c)

Figure 3.4: Feature Detection Example. Example feature detection of Good Features and SIFT features. Note that SIFT removes features detected along predominant edges.

gyroscope bias($\delta b^b$) errors [38]:

$$\delta x = \begin{bmatrix} \delta p^n \\ \delta v^n \\ \psi \\ \delta a^b \\ \delta b^b \end{bmatrix}_{1x15} \tag{3.1}$$

Propagation of the navigation state error uses the kinematic equations of motion. Landmark positions augment this state vector. In this research, 10 feature locations were tracked, and the complete state vector consisted of 45 elements.

The new image update replaces SIFT features with Good Features. Image matching is now accomplished with a correlation coefficient computation, rather than a simple Euclidean distance (see Sections 2.5.1 and 2.5.2). Figure 3.6 highlights the differences in each algorithm. As previously discussed, these features are less invariant to rotational and scaling changes, and techniques to aide the landmark matching will be discussed in the next section.

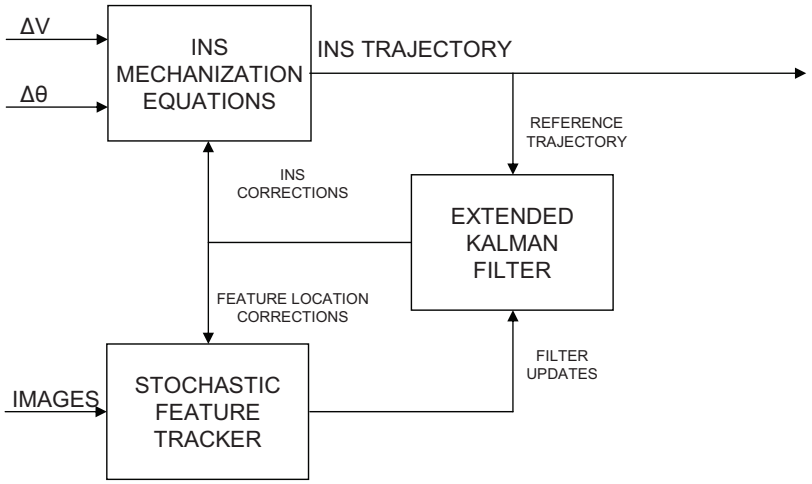



Figure 3.5: Image-aided Kalman Filter Block Diagram [38]. The image-aided Kalman filter is decomposition into the extended Kalman filter, the inertial mechanization (inertial update), and the stochastic feature tracker (image update).

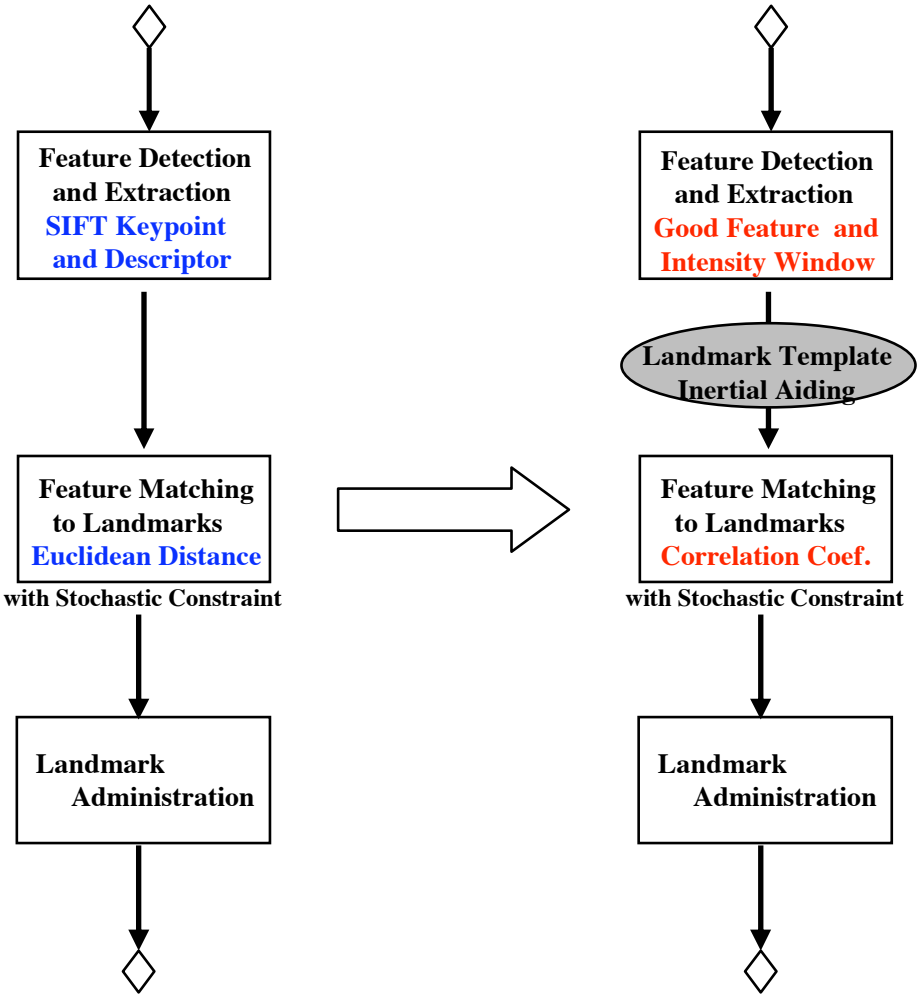



Figure 3.6: New Image Update. This figure summarizes the modification made to the original extended Kalman filter. The computationally intensive SIFT features are replaced with low-level Good Features with inertial aiding of descriptor matching.

Figure 3.7 shows the matching process between two image frames. In previous research, the predictive transformation involved the feature location. The current two-sigma uncertainty is searched for the strongest match, and this reduced search space is called the stochastic constraint. SIFT features are more invariant to arbitrary camera motion and primarily benefitted from the number of comparisons necessary to find a match. For intensity window matches, an added benefit is the reduction in false matches. False matches are common in indoor intensity matching because lights and doorways have common geometry and little texture. False matches incorporated into the filter are detrimental to the navigation solution.

*3.2.1 Tuning the Feature Matching.* A threshold must be established for the correlation coefficient template matching. Figure 3.8 shows the tradeoff in accuracy for different thresholds. If the threshold is set to a high value, the feature matching will reject features misaligned by a few pixels. Conversely, a low threshold with allow for too many erroneous matches. Thus, there is a tradeoff between pixel accuracy and invariance to small 6DoF motion changes, such as the matching between binocular camera disparity. A threshold of 93 percent was empirically found to produce a good match without significant pixel misalignment. Also, the selected threshold performed well over the small affine warping between the stereo cameras.

## *3.3 Deep Integration of Inertial and Imaging Sensors*

In addition to the prediction of location, this research predicts the feature descriptor in the next camera frame. This is an effort to compensate for the lack of invariance of the intensity descriptor. Descriptor aiding assists matching in the presence of rotation and 6DoF motion changes. This descriptor aiding assumes, despite changes in the strength, features can be repeatedly detected in the presence of the motion. In other words, successful aiding requires that features be detectable over the same camera motion.

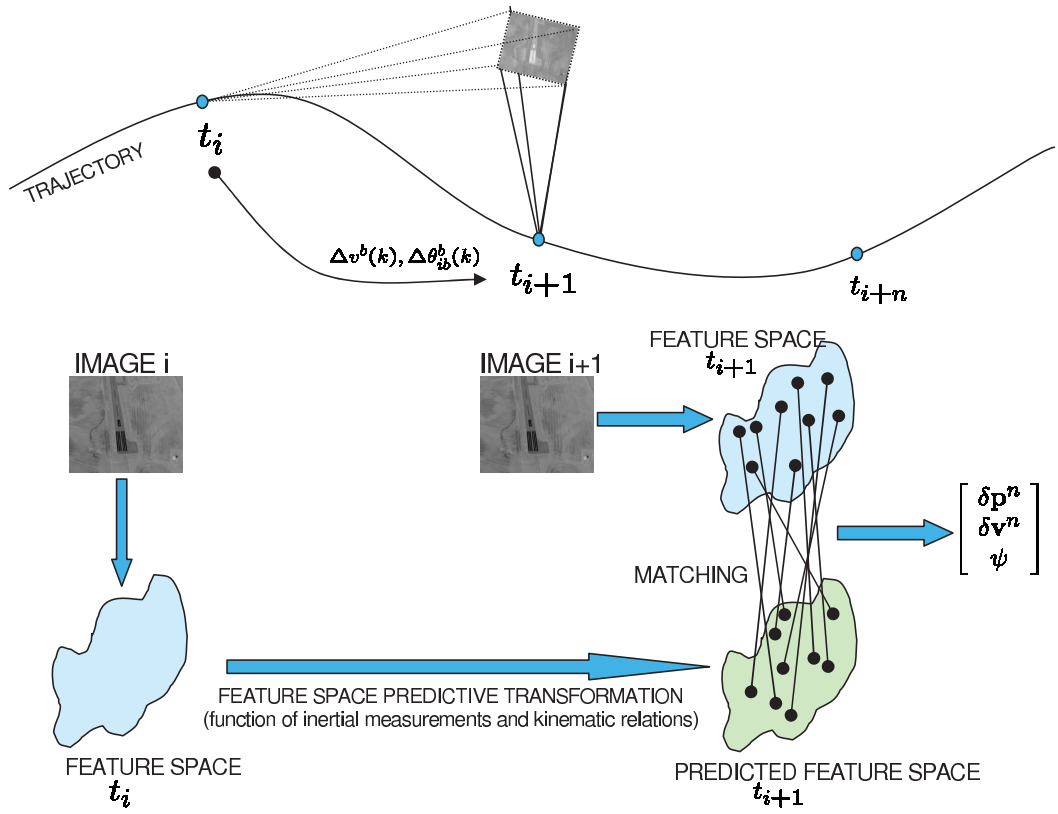



Figure 3.7: Image-aided Kalman Filter Diagram [38]. The image-aided Kalman filter uses images to improve the state estimate and state estimation to improve the feature tracking. This research implements a transformation of the descriptor and uses a lower level feature detection and extraction [38].

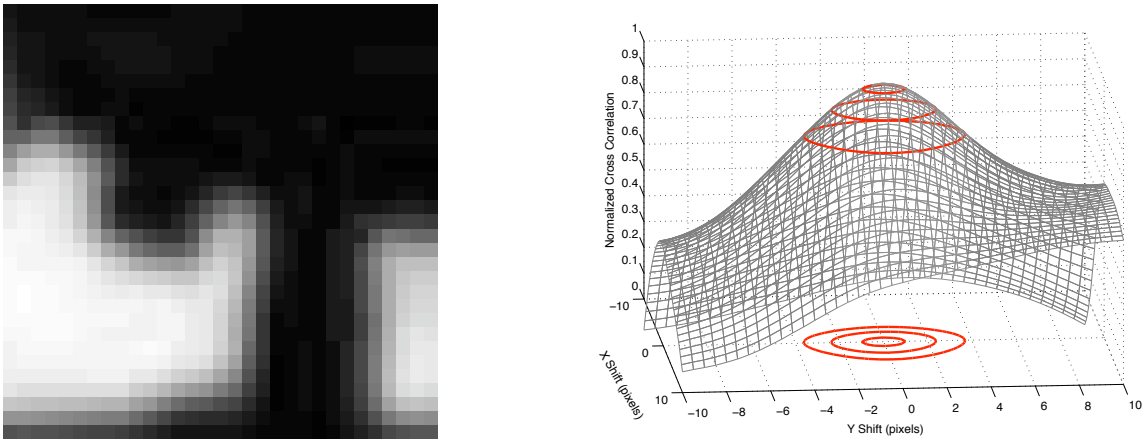
Aiding will allow for longer landmark tracking and has two primary benefits. First, tracking a landmark longer will increase the certainty of the location and the contribution to the navigation solution ($\delta x$) (see Section 2.6.2.2). Also, landmark initialization can be an expensive operation in the filter requiring statistical calculation of the influence matrices (see Section 2.8.2). Longer tracking means less landmark initializations.

In the next sections, rotational and 6DoF descriptor aiding are introduced.

*3.3.1 Rotational Descriptor Aiding.* Rotational aiding removes rotations along the camera frame's z-axis from the time the initial template was captured to the current orientation. The image rotation can be determined from the z-dimension of the camera to navigation frame direction cosine matricies ($C_n^{c1}$,$C_n^{c2}$) at each time epoch. Each pixel is transformed according to the following relation:

$$p' = \left[C_n^{c2} C_{c1}^n\right]_z p \tag{3.2}$$

where $p$ is the set of descriptor pixels in the previous frame, and $p'$ is the pixels transformed into the current frame. Bilinear interpolation of pixel values is used for




(a) (b)

Figure 3.8: Template Shift Analysis. (a) shows a example Good Feature template. Shifted versions of the template are correlated with the original template. Example thresholds of 98, 90, and 80 percent are plotted in (b).

the uniform sampling lattice of the new frame. Figure 3.9 illustrates the rotational descriptor aiding.

The feature matching threshold affects when rotational aiding will have any benefit. A rotational descriptor experiment was conducted to demonstrate when aiding would benefit matching. In the experiment, different sample templates where rotationally transformed and compared to the original template. Figure 3.10 shows

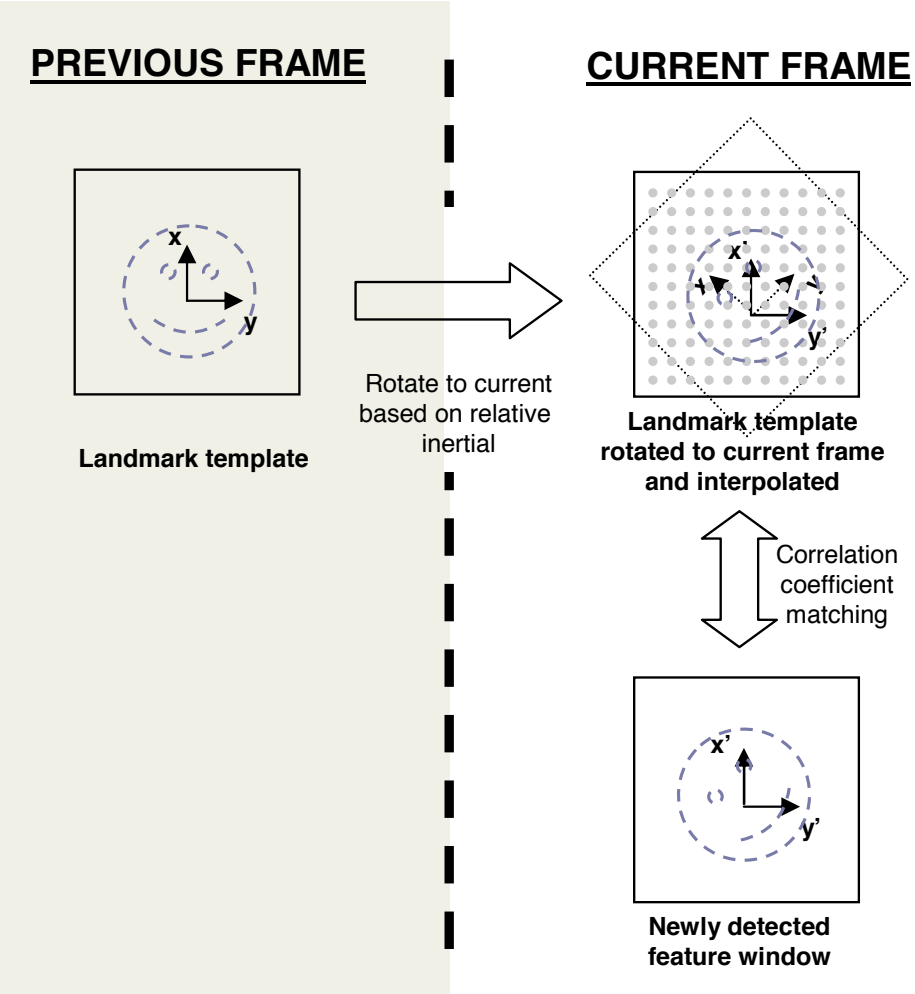



Figure 3.9: Rotation Aiding Example. The tracked landmark is rotated into the new frame by using z-dimension of the rotation between the previous and current frame. Each pixel is transformed through the rotation and then resampled to fit the current frames sampling lattice.

of rotation increases. As the matching threshold increases, the tolerance for rotation decreases. For the matching threshold of 93 percent, matching should be unaffected by rotations less than 5 degrees. Level flight profiles, consisting of translation only should not benefit significantly from the rotational aiding.

Many of the feature descriptors determined by the Good Features algorithm are partially invariant to scale changes. Thus, rotational aiding alone can benefit during a scale change in the camera. When these conditions do not exist or over larger motions, perspective warping, called 6DoF descriptor aiding, is used.

*3.3.2 Six Degree-of-Freedom Motion Descriptor Aiding.* The second type of descriptor aiding investigated in this research was 6DoF aiding. Good Feature detection is strong and repeatable over scaling situations, however the intensity descriptor is not always invariant. The previous SIFT algorithm's descriptor provides some invariance to severe 6DoF camera motion changes (40 to 70 degrees) [18], matching approximately half the descriptors in these cases. In an effort to achieve similar performance, the concept of 6DoF motion descriptor aiding is introduced.

This research proposes a homographic transform based on inertial information to provide 6DoF camera motion invariance. In the image plane, 6DoF motion is per-

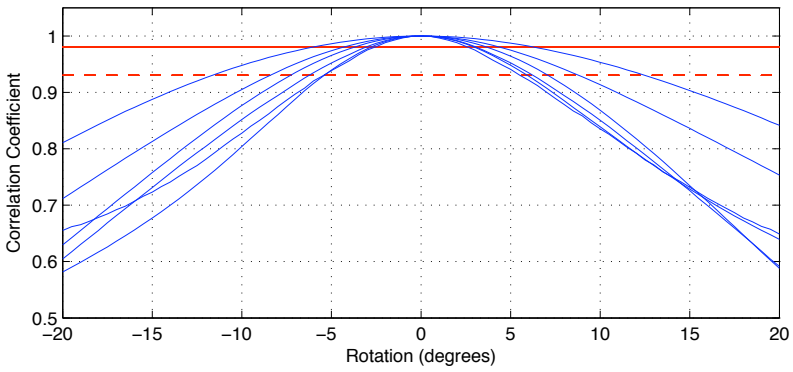
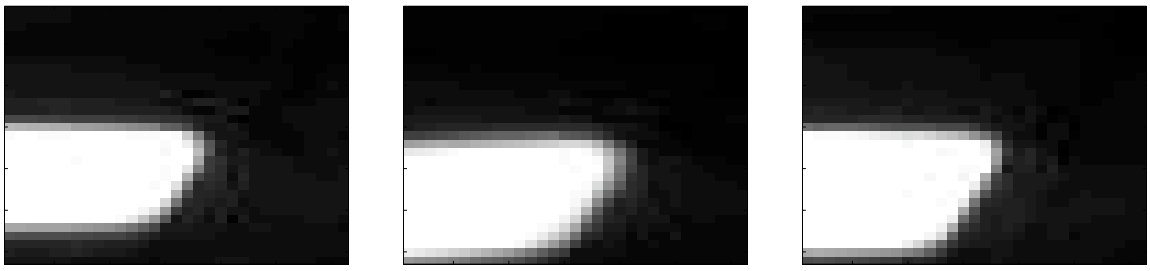



Figure 3.10: Correlation Analysis for Template Rotation. Templates are rotated then self-correlated to demonstrate the effect of rotation on matching. As the matching threshold is increased, the tolerance for rotation decreases. Threshold lines are shown for 98 and 93 percent correlation.

ceived as a perspective projection. A homography is a special type of perspective projection for a planar surface in the scene. The homography is computed according to Equation (2.18) and involves the camera rotation, translation, and calibration as well as the normal vector and distance to the plane. In this research, only ceiling features are considered, and the planar normal is assumed to be pointing down in the navigation frame. The distance to the plane is determined by computing a cross product of the normal vector and line-of-sight vector to the feature. Each pixel from the descriptor's intensity template is mapped into the current frame (see Section 2.3.5). Pixels are then resampled according to the current frame's sampling lattice using bilinear interpolation. Figure 3.11 shows a representative example of 6DoF motion aiding from the experiments discussed in the next chapter. The tracked landmark template is shown on the left. The feature on the right is the candidate match in the next frame. Using the homographic transform, the tracked landmark is warped into the current frame resulting in the center image. This warped template is properly matched to the candidate feature in the current frame.

In the next section, a monocular initialization technique is introduced to further minimize the computational complexity of the image-aided Kalman filter.



(a) (b) (c)

Figure 3.11: Six Degree-of-Freedom Motion Aiding Example. The image in (a) represents the unaided landmark template being matched to the feature window in (c) (the landmark from a new camera pose). The landmark template in (b) has been warped using a homography defined by inertial information.

## *3.4 Monocular Landmark Initialization*

In initial testing, binocular landmark initialization is assumed to eliminate the need for *a priori* information. However, monocular initialization of landmarks is an important requirement for the MAV application due to limited processing and space on the aircraft. For monocular initialization in this research, a landmark depth is determined by a statistical distribution. The mean of the distribution is a guess of the maximum distance expected, and the distribution is given a high uncertainty. The high mean means the IAKF will predict smaller motions between frames based on inertial movement, and the high uncertainty increases the correspondence search space for landmarks. This differs from monocular initialization presented in the previous research [38] because the depth does not depend on the slant range to a defined reference terrain (see Section 2.8.2). Instead, the depth is static and adjusted as the filter matches the feature in the next frame.

This chapter selected a low-level feature transformation and introduced techniques for feature descriptor aiding. In the next chapter, indoor flight experiments are used to validate this deeply-integrated feature tracker.

# IV. Results

This chapter presents testing results comparing the deeply-integrated tracking algorithm to the previous SIFT-based tracker. First, the computational costs for each algorithm are compared to show the speed improvement. Next, three indoor flight experiments are conducted to exercise the Good Features transformation with different types of inertial aiding. These results are compared with the previous filter developed in the SIFT-based research [38] to characterize accuracy differences. The first test mirrors the indoor experiment presented in the previous research. The next experiment introduces rotation while moving down the hallway as well as a banked turn. This experiment exercises the benefits of inertial aiding. Finally, the last experiment demonstrates image-aided Kalman filter (IAKF) monocular and binocular estimation during an aircraft in hover within the Air Force Research Laboratory's (AFRL) micro air vehicle (MAV) laboratory. This MAV laboratory allows for a precise truth trajectory to compare each algorithm's estimated trajectory.

## *4.1 Computational Cost Analysis*

A primary goal of this research was to reduce the computational cost of the feature tracking in the IAKF. Table 4.1 shows the processing speeds of each algorithm for detection/extraction and matching. Besides aiding techniques, all other IAKF stochastic feature tracking parameters were identical. For SIFT matching, the dot product method (see Section 2.5.1) was used for the 128 element descriptors. A window size of 31x31 pixels was used for the Good Features descriptor, and the descriptor was matched using normalized cross correlation. Algorithms were implemented in the C programming language and run on a 1.06 GHz Pentium M system with 2 GB of memory.

Because of the simplicity in detection and extraction, Good Features has a 22 times speed increase over the scale-invariant feature transform (SIFT). Because of its one dimensional aspect, SIFT descriptor matching is twice as fast as the Good Features matching. Assuming that the number of matching attempts remains the

same on average, Good Features has 6 seconds to perform aiding techniques to improve the robustness of feature matching. Although implemented in MATLAB and not C, the average length of time for these transforms were 0.01 seconds for rotation and 0.25 seconds for 6DoF motion aiding. At a maximum, this aiding occurs once for each landmark during an update. Overall, the new low-level tracking algorithm is 12 times less expensive than the previous algorithm. Next, the experimental setup and truth reference system used in the indoor flight experiments are introduced.

## *4.2 Hardware Overview*

Feature tracking performance is evaluated in the context of the image-aided inertial Kalman filter. Real imaging situations are difficult to accurately model in simulation, so real image sequences were captured used in experiments. The hardware used for the flight experiments is reviewed in this section. The experimental test setup used two cameras and one commercial grade inertial measurement unit (IMU). Images were captured at an average rate of 2.3 Hertz. For the last indoor flight experiment, the Vicon motion capture system provided a truth reference trajectory.

*4.2.1 Experimental Test Setup.* Data was collected using two PixeLINK [30] cameras and one commercial-grade MIDG [23] inertial measurement unit. The PixelLINK camera has a resolution of 1024x1280 pixels. Table 4.2 gives the specifications for the MIDG IMU. During experiments, it was estimated that the MIDG IMU can provide an accurate positional solution for approximately 10 seconds without aiding.

Table 4.1: Computational Cost Analysis. Average computation costs are compared for SIFT and Good Features detection/extraction and matching.

| | SIFT | Good Features |
|---|---|---|
| Detection/Extraction | 6.70 sec | 0.30 sec |
| Matching | 0.10 sec | 0.21 sec |

Table 4.2: MIDG II Specification Summary. The MIDG II is a combined GPS/IMU unit that can operate up to 50hz. For this research, the GPS positional output was ignored. Parameters marked with an asterisk are estimated [38].

| Parameter (units) | Value |
|---|---|
| Sampling interval (ms) | 20 |
| Gyro bias sigma (deg/hr) | 1800 |
| Gyro bias time constant (hr) | 2* |
| Angular random walk (deg/$\sqrt{hr}$) | 2.23 |
| Gyro scalefactor sigma (PPM) | 10000 |
| Accel. bias sigma (m/$s^2$) | 0.196 |
| Accel. bias time constant (hr) | 2* |
| Velocity random walk (m/s/$\sqrt{hr}$) | 0.261 |
| Accel. scalefactor sigma (PPM) | 10000 |

Figure 4.1 shows the complete data collection test setup. After collection, the data was post-processed in the image-aided navigation filter using MATLAB.

*4.2.2 Vicon Motion Capture System.* The Vicon motion capture system serves as a truth reference system for the indoor hover flight test in the Air Force Research Laboratory's micro air vehicle (MAV) laboratory. The system provides high-rate, accurate three-dimensional positional estimate using 36 cameras. Reflective visual markers are placed on the object for the cameras to observe. The final trajectory estimate of location and attitude is determined by Vicon's IQ software [40].

Next, the flight experiments are examined to demonstrate the performance of the new stochastic tracker in comparison to previous tracker results.

## *4.3 Indoor Flight Experiments*

Three experiments exercise the previous SIFT-based and Good Features tracking algorithms. In each experiment, a stationary alignment update is applied at the beginning of the run to estimate the biases in the gyroscopes and accelerometers. After each run, the estimated horizontal and vertical trajectory as well as the number of initialized landmarks were recorded. The same filter parameters were used in each run, only the feature tracking aspects were changed. The final trajectory estimate

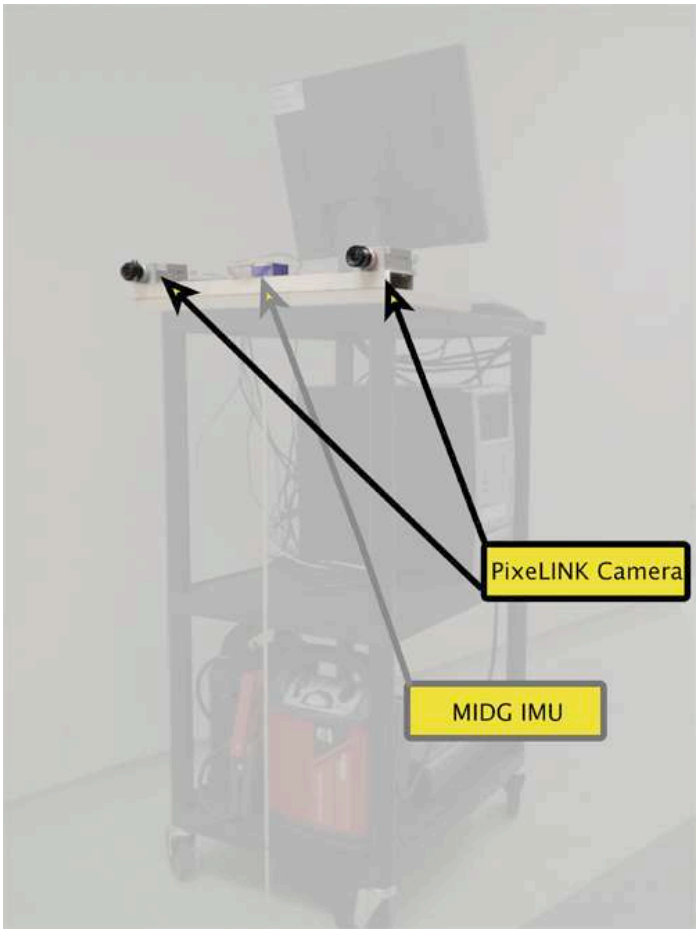



Figure 4.1: Experimental Setup. Two PixelLINK cameras and one MIDG inertial measurement unit are mounted on a sensor bar and moved throughout the hallway. Care was taken to keep the IMU at the same height through the flight profile.

gives an indication of the accuracy of the IAKF estimation with each algorithm. The number of landmarks initialized is an indicator of how long landmarks were tracked and the robustness of the feature tracking algorithm. Landmarks tracked longer have less uncertainty and more influence on the navigational estimate. Also, landmark initialization involves computation of uncertainty statistics which slows down the overall image update.

For the first two experiments, 6DoF aiding was accomplished by assuming features at a predetermined height were on the ceiling, and thus the planar normal pointed down. Also, landmarks are initialized with binocular techniques.

For initial flight testing, the IAKF was run on a monocular platform and flown in the MAV lab at AFRL. The final experiment demonstrates the IAKF's performance during the hover condition using the the SIFT-based and Good Features algorithms. Monocular initialization is also compared to binocular initialization. The MAV lab provides an accurate position and attitude truth reference system for the flight trajectory, and a statistical analysis of the each tracker's output is presented to further quantify performance.

*4.3.1 Hallway Experiment.* The first experiment follows a closed-loop path in a hallway. The experimental setup was kept at a constant vertical height throughout the flight path. Figure 4.2(a) shows the closed path for each estimated trajectory. The SIFT-based estimated trajectory serves as the previous research baseline. With no stochastic correspondence search constraint, the Good Features estimated trajectory quickly diverges. The divergence is due to false matches entering the filter and corrupting the trajectory estimate. With the stochastic constraint of the search space, the drift is constrained and the filter is able to produce an accurate trajectory estimate. In fact, the inertial-aided, low-level tracker generally performs with an accuracy greater than the SIFT tracker in this experiment.

Table 4.3 shows the number of landmarks initialized for each algorithm. Thirty percent more landmarks were initialized by the low-level tracker during the exper-

Table 4.3: Hallway Experiment Landmarks Initialized. A lower number of landmarks initialized indicates higher feature tracking performance.

| | Landmarks initialized |
|---|---|
| SIFT | 806 |
| Good Features | 1066 |
| Rotational Aiding | 1016 |
| 6DoF Aiding Alone | 1040 |
| 6DoF and Rotation | 1014 |

iment. This increase indicates that SIFT features are still more robust than the inertial-aided, low-level features but does not necessarily indicate a slower image update. Typically in MATLAB, the binocular landmark initialization took 0.75 seconds (depending on the number of features and their distribution in the scene). Over the entire flight profile, the SIFT-based tracker would have a speed advantage of 195 seconds considering only the landmark initialization. However, there were also a total of 1688 image updates each requiring a feature transformation to be performed. Individual low-level feature transforms had a speed savings of 6 seconds. The combined transformation savings was 50 times the additional cost of landmark initialization over the run. Thus, the deeply-integrated tracker performs considerably faster and reduces computational complexity.

Results vary in the trajectories for a few possible of reasons. In the southeast corner of the horizontal trajectory, a saturation condition occurs in the camera resulting in feature starvation. Heading deviations in this corner are likely attributed to this saturation. Furthermore, the features tracked in each trajectory differ because of staggered landmark initialization events. An important observation to make is the estimated trajectories are consistent during the run. This demonstrates successful constraint of the inertial drift. Typically, the commercial-grade inertial sensor would drift after approximately 10 seconds.

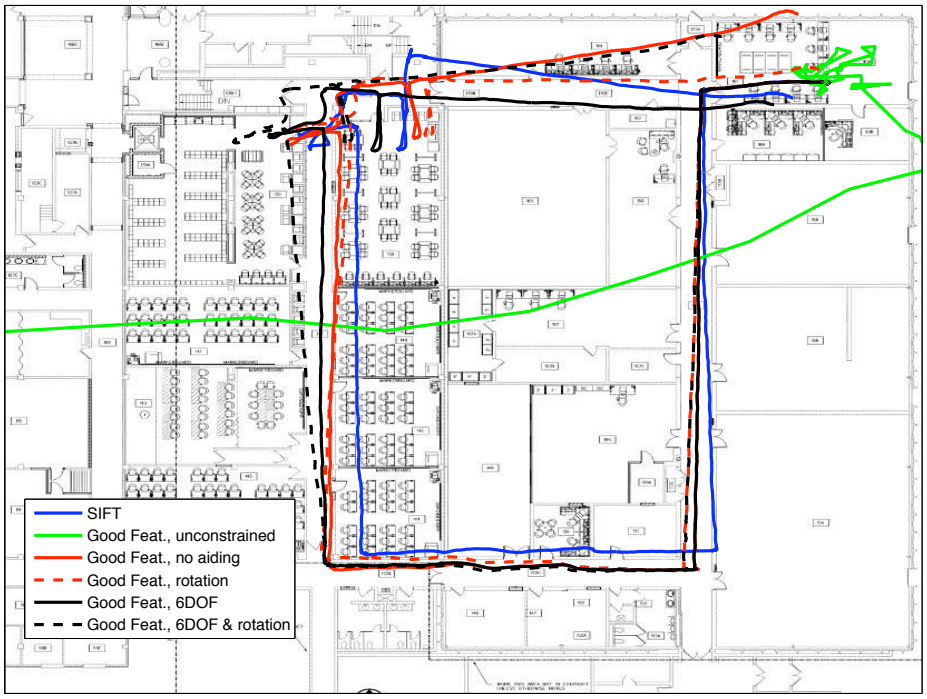
The unaided Good Features estimated trajectory is as good or better than the previous SIFT-based estimated trajectory. Aiding was applied to increase the robustness of matches and reduce the number of landmarks initialized. Three aiding

combinations were testing in this experiment: rotation only, 6DoF only, and a combination of the two techniques. The 6DoF aiding trajectory performed well but diverged in the vertical trajectory. This is shown in Figure 4.2(b). The divergence can be attributed to only keeping features that are tracked on the ceiling. Notably, 6DoF aiding did not significantly improve the number of features initialized or the overall trajectory in this experiment. Further investigation found that only 30 percent of the 6DoF-aiding attempts were able to produce an improved normalized cross-correlation and match in the experiments. This indicates that either a violation of the planar template assumption or improper estimation of the planar normal vector.

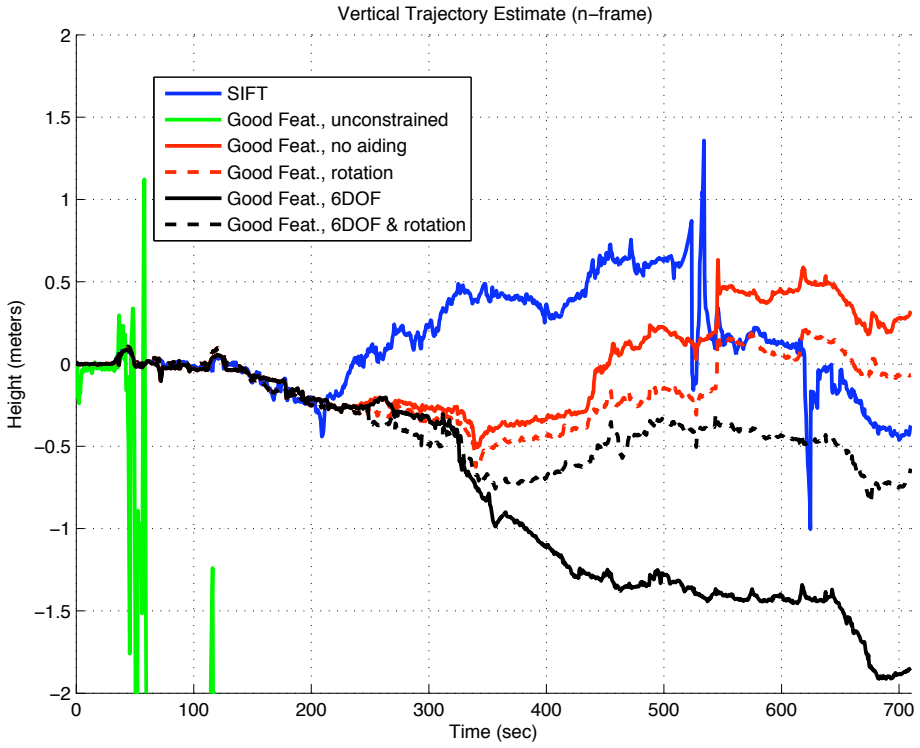
This flight experiment did not have enough rotation to emphasize the benefits of inertial descriptor aiding. The next flight experiment focuses on showing these benefits by introducing a rotation into the flight profile.

*4.3.2 Severe Motion, Hallway Experiment.* In the second experiment, the flight profile moved straight down a hallway with a 30 degree banking oscillation. A banked turn is executed at the end of the hallway, and the aircraft proceeds down that hallway. Again, the true vertical trajectory remains constant throughout the data collection. This experiment is meant to demonstrate the need for rotational aiding depending on the flight profile. Figure 4.3 shows the results of the experiment for each algorithm.

The Good Features trajectory estimate provides similar horizontal accuracy but considerably less accuracy in the vertical trajectory. Rotational aiding provides a more accurate horizontal estimate and drifts slightly less in the vertical trajectory estimate. 6DoF-aiding further reduces the vertical drift but performs poorly in the banked turn at the end of the hallway. Table 4.4 shows the landmarks initialized by each algorithm. In this experiment, aiding techniques significantly improved the number landmarks initialized for Good Features tracking. This indicates an increase in the robustness of the feature descriptor occurred.

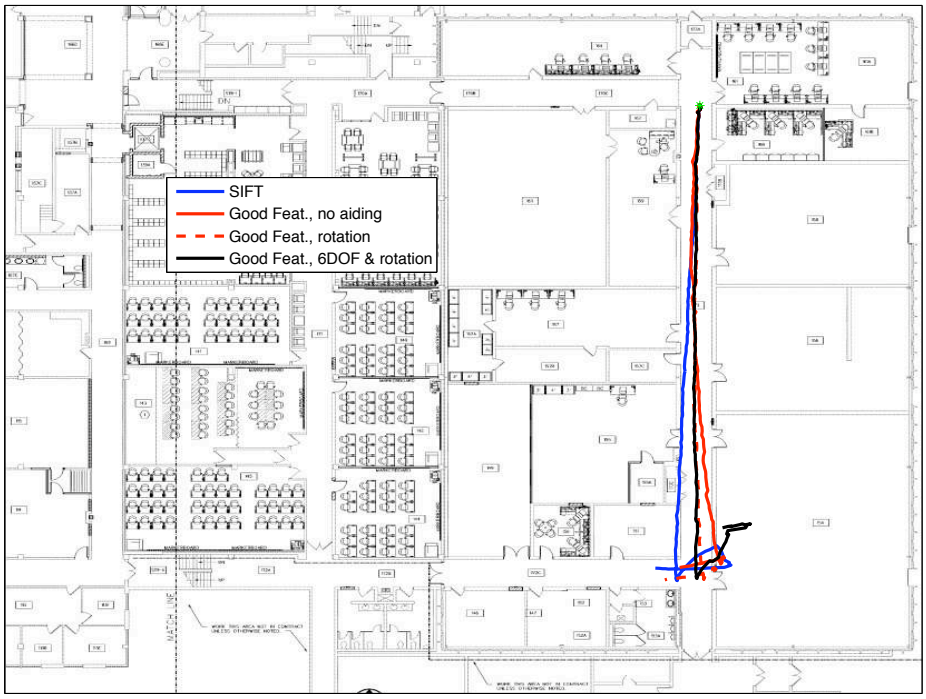



(a)

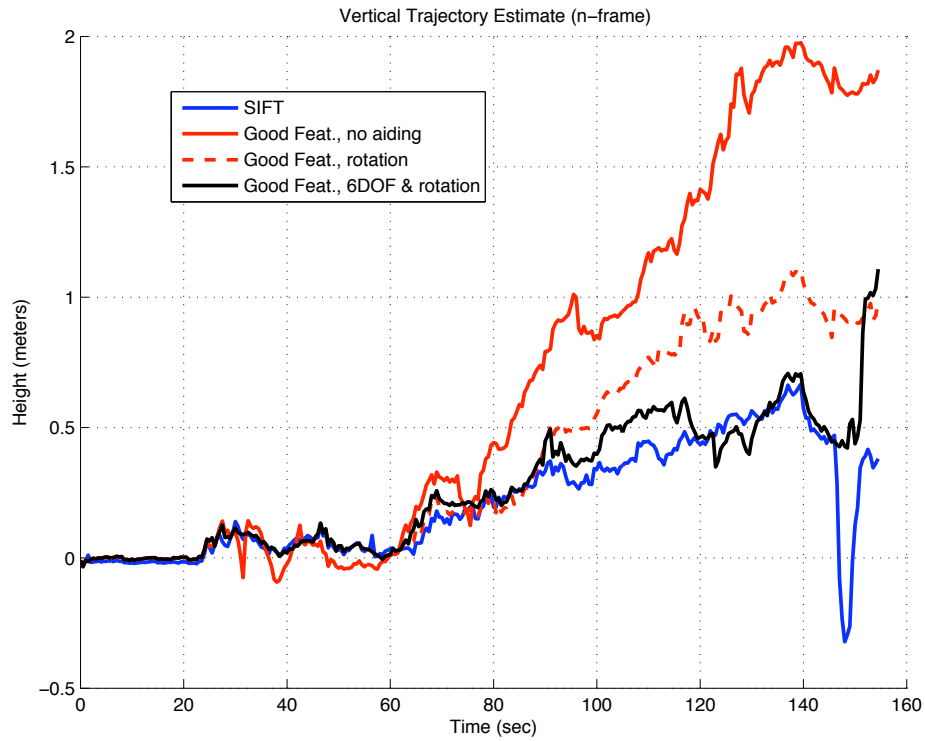



(b)

Figure 4.2: Hallway Experiment Estimated Trajectories. The overall position estimate is observed over a 10 minute closed loop flight profile. Note that the true vertical trajectory is zero for the duration of the flight.




(a)




(b)

Figure 4.3: Severe Motion Hallway Experiment Estimated Trajectories. In this experiment, the camera was rotated left and right during a straight path down a hallway to clearly demonstrated the benefit of descriptor aiding for low-level features.

Table 4.4: Severe Motion Hallway Experiment Landmarks Initialized. A lower number of landmarks initialized indicates higher feature tracking performance.

| | Landmarks initialized |
|---|---|
| SIFT | 184 |
| Good Features | 338 |
| Rotational Aiding | 260 |
| 6DoF and Rotation | 274 |

In the next section, aiding techniques are not used. Instead, monocular and binocular feature initialization is analyzed for an indoor hover condition to further reduces computational complexity.

*4.3.3 Indoor Flight Facility Hover Experiment.* The Air Force Research Laboratory (AFRL) MAV lab provided the environment for the final experiment. The goal of the experiment was to prove the IAKF could provide a stable and accurate solution during a vehicle hover. The Vicon flight motion capture system provided a position and attitude truth reference for this experiment. Trajectories generated by the previous and new versions of the IAKF were analyzed. Binocular and monocular initializations are examined, but aiding was not used. At the beginning of the run, an alignment update was performed (without movement) for 30 seconds. Next, the test platform was lifted to a stable hover and finally brought down for a landing.

Table 4.5 shows the landmarks initialized during the experiment. In either feature transformation, monocular camera initialization caused a severe increase in the number of landmarks initialized. The increase for the Good Features algorithm was more severe and increased by three times the number of landmarks using binocular initialization.

Figure 4.4 shows the horizontal trajectory of binocular and monocular simulations. The truth provided from the Vicon system is shown in black. Located at the top of each figure is the flat wall of the facility. Figure 3.4(c) shows an image of the facility wall. A zoomed view of the estimated trajectory is shown on the right side of the figure. The binocular estimated trajectory errors are on the order of tenths of

Table 4.5: MAV Lab Experiment Landmarks Initialized. A lower number of landmarks initialized indicates higher feature tracking performance.

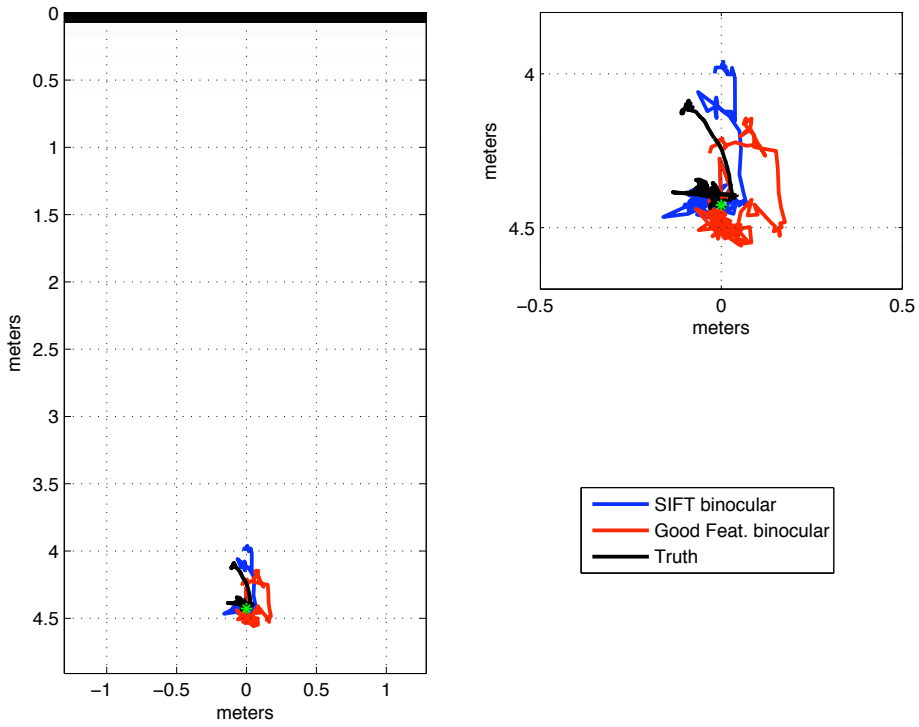
| | Landmarks initialized |
|---|---|
| SIFT binocular | 36 |
| SIFT monocular | 68 |
| Good Features binocular | 30 |
| Good Features monocular | 90 |

meters. The monocular estimated trajectory still followed the truth trajectory but with less accuracy than the binocular estimation. Still the errors are on the order of tenths of meters.

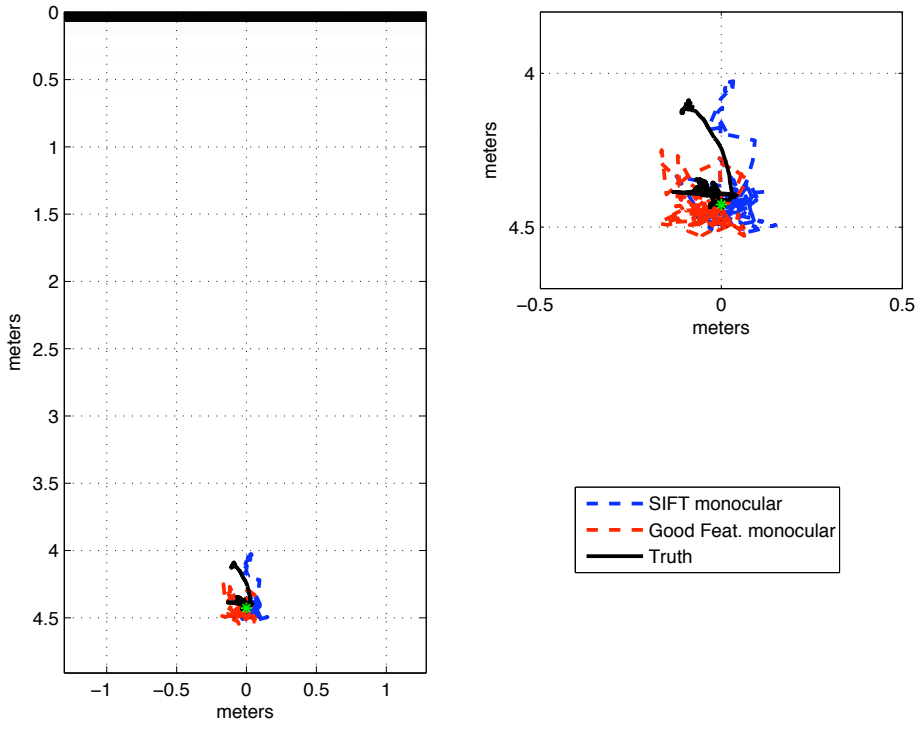
Figure 4.5 shows the complete position and attitude trajectory for the binocular and monocular runs. Notice that in every case, there is trouble tracking the easting position that is perpendicular to the facility wall. This is a result of a lack of observability on the scale necessary to produce precise results. However, the the errors are bound to tenths of a meter. In the attitude estimation, there is an unobservability in the yaw dimension. The error is most predominant in the binocular algorithms. This difference can be attributed to the dominant horizontal trends in the image, and few vertical trends in the image (see Figure 3.4(c)). Overall, these results show that a stable estimated hover trajectory can be achieved with each of these algorithms.

Further analysis focuses on the uncertainty of each of the estimated trajectories. For a properly functioning IAKF, the filter should predict the true trajectory within one-standard deviation on average for an ensemble of runs. This research has one sample run from the ensemble and is not guaranteed to fall within these uncertainty bounds. Also, the EKF is know to be a statistically biased estimator, and this could contribute to observed biases (see Section 2.6.2.2).

Figure 4.6 shows SIFT binocular trajectory with uncertainty. This algorithm's performance serves as the baseline for the rest of the algorithms. The statistical uncertainty accurately captures the true trajectory in position and attitude. There was a slight bias in yaw estimate, but the estimate still follows the trend of the

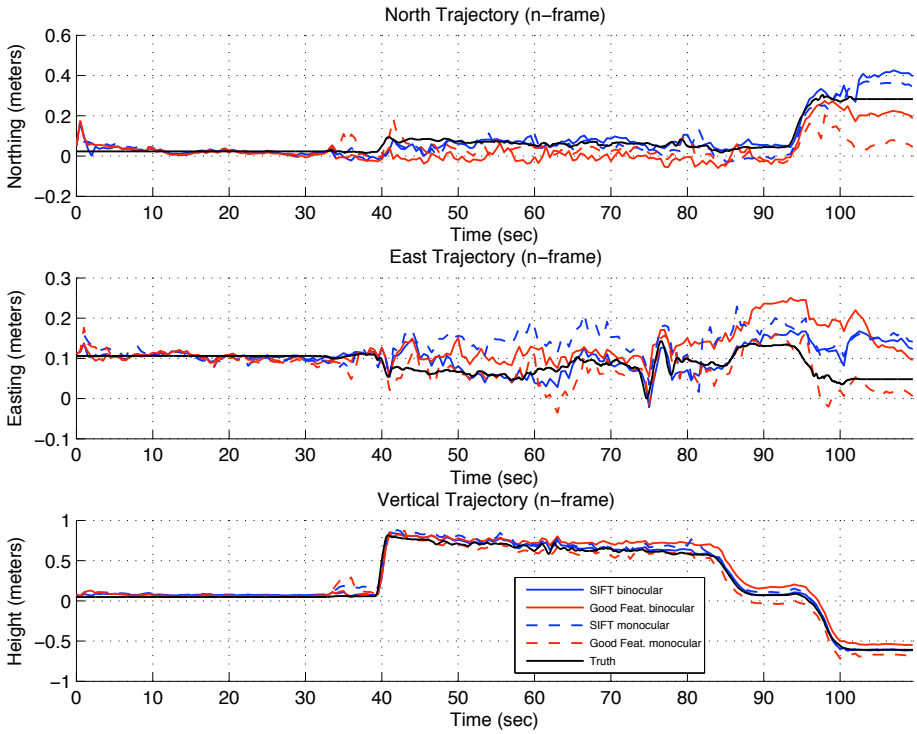



(a)

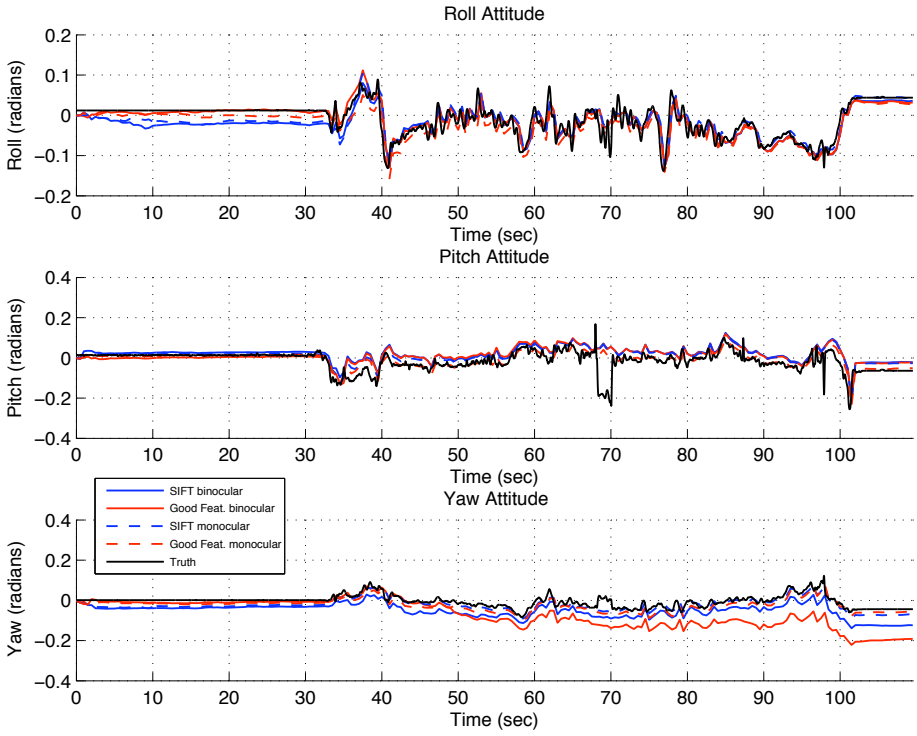



(b)

Figure 4.4: MAV Lab Horizontal Estimated Trajectories. The binocular estimated trajectory, shown in (a), closely tracks the true trajectory. Monocular results, shown in (b), give slightly less precision. In either case, the positional error is on the order of tenths of meters.




(a)




(b)

Figure 4.5: MAV Lab Full Estimated Trajectories. The trajectories show that beside the low observable easting trajectory, the algorithms perform well constraining error to tenths of meters.

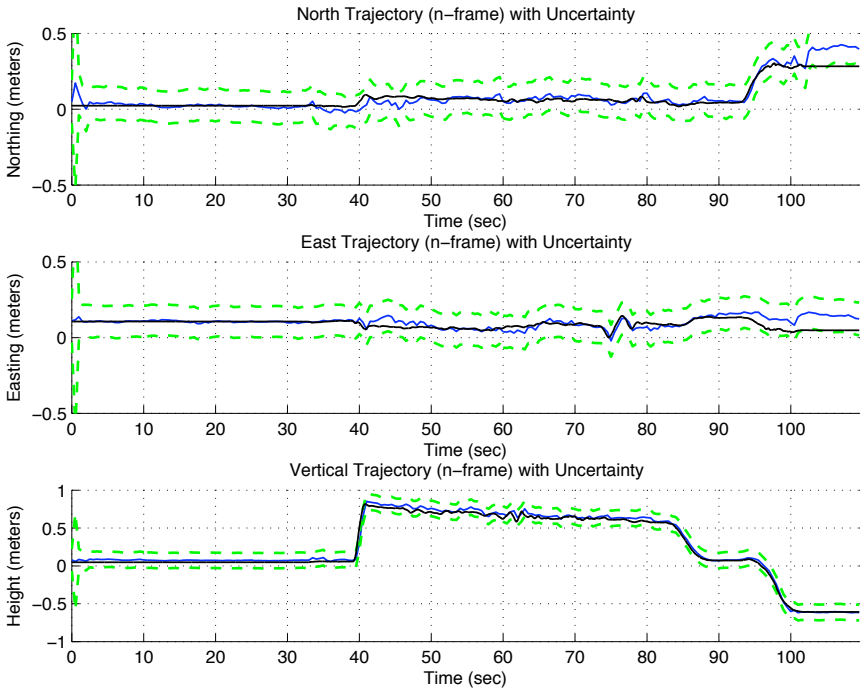
truth trajectory. In the pitch truth trajectory at 70 seconds, there is a jump in the data. This jump is likely do to a reflection on the experimental setup causing the Vicon visual reference system to improperly estimate the trajectory. This was the only significant jump observed in the truth data. Overall, the binocular SIFT tracker performed within the statistical uncertainty.

Figure 4.7 shows the Good Features binocular trajectory estimate. This tracker has comparable performance to the estimated binocular SIFT trajectory with one exception. Although a constant bias is not observed, there was drift in the estimated yaw not captured by the one-sigma uncertainty. The divergence was approximately 10 degrees at the end of the run, but stabilized at the end of the flight. This drift begins when the vehicle is raised to a height of one meter and was likely caused by a poorly matched feature due to a reflection on the wall. With the exception of the yaw, the binocular Good Features estimated trajectory accurately captures the truth trajectory, and the tracker performed well.

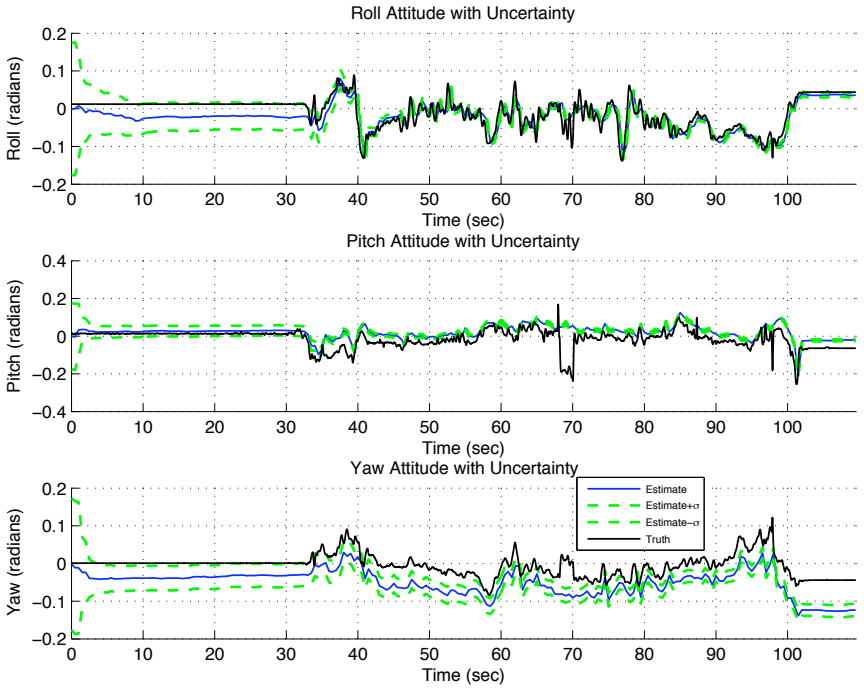
Figure 4.8 shows the monocular SIFT trajectory estimate. The bias observed in the binocular SIFT trajectory estimate is not present, and the filter performs within the statistical uncertainty. As noted previously, the monocular initialization's accuracy was retained at the cost of additional feature initializations.

Figure 4.9 shows the monocular Good Features trajectory estimate. Again, the tracker captures the truth trajectory in the one-sigma uncertainty bound. The north trajectory estimate did start to diverge after 100 seconds. This divergence happens as the vehicle approached the floor of the facility. The most likely cause of the divergence is the loss of tracked features. With the exception to landing, the monocular Good Features trajectory estimate performed within the statistical uncertainty and equally as well as the binocular SIFT estimation.

For a final analysis of the MAV Lab experiment, root-sum-squared (RSS) errors of position and attitude of each estimated trajectory were computed. The horizontal and vertical RSS errors are plotted in Figures 4.10 and 4.11. The binocular SIFT

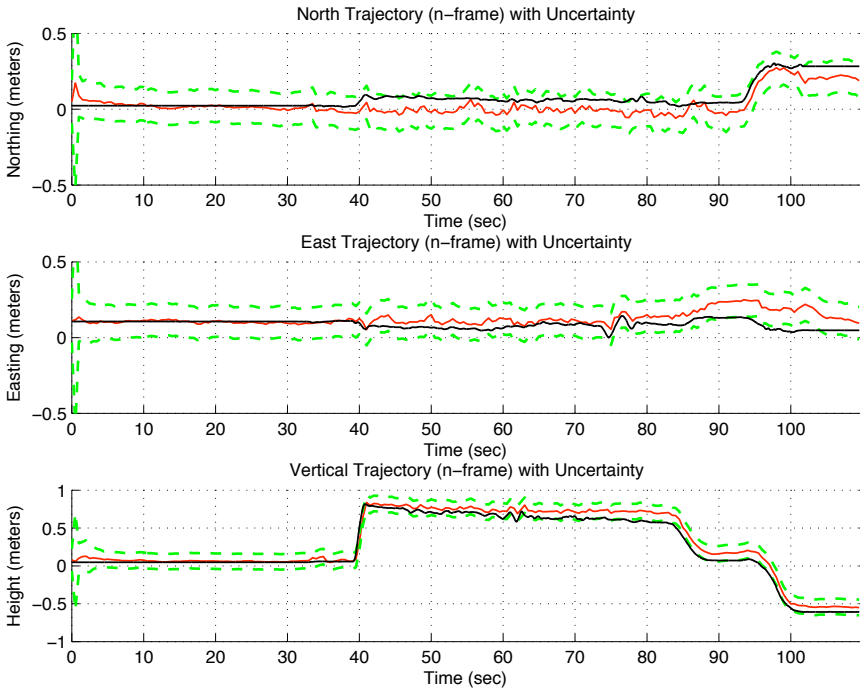



(a)

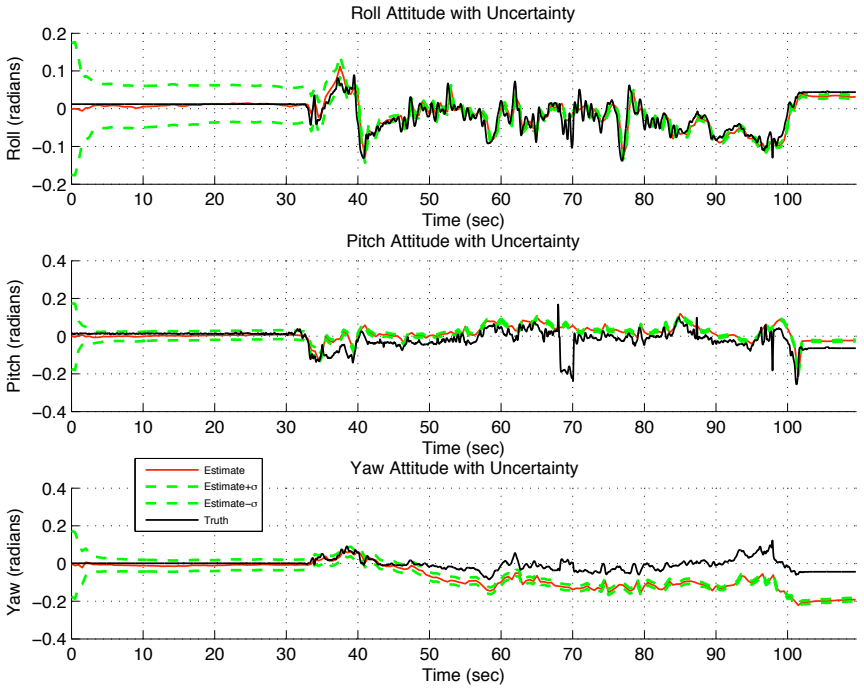



(b)

Figure 4.6: MAV Lab SIFT Binocular Estimated Trajectory with Uncertainty. The binocular SIFT estimated trajectory is plotted with one-sigma uncertainty and the truth trajectory provided by the Vicon vision system.

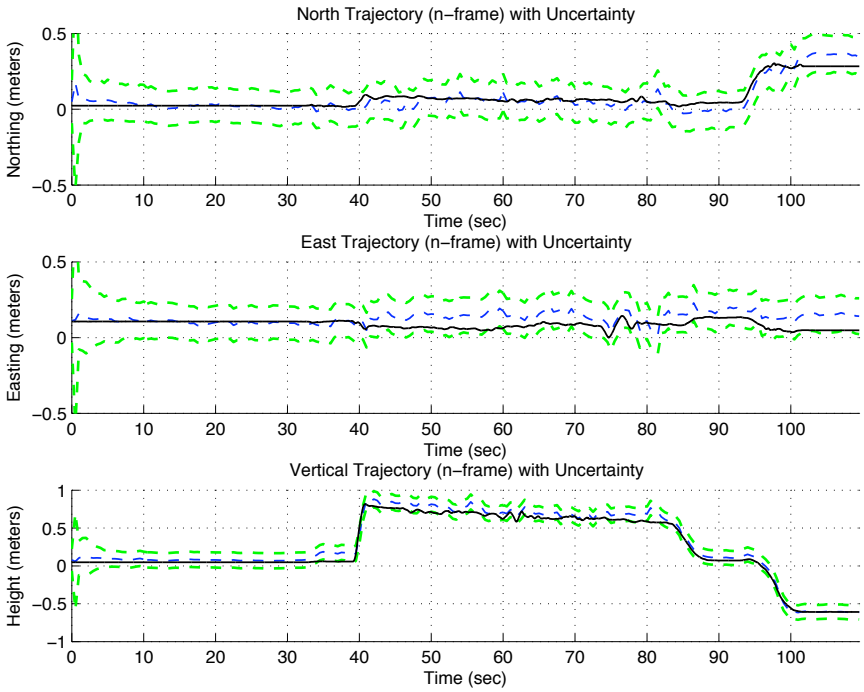



(a)

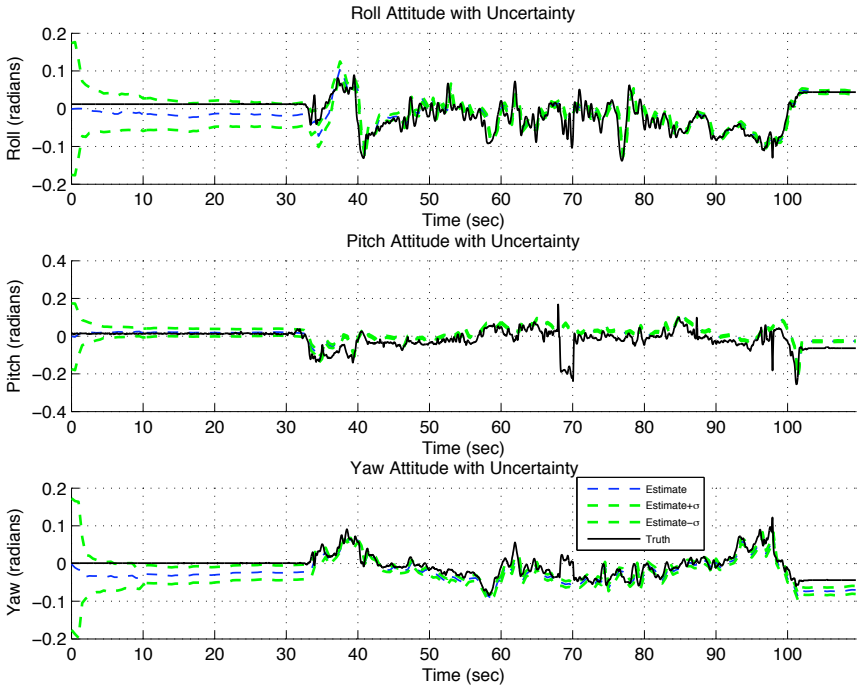



(b)

Figure 4.7: MAV Lab Good Features Binocular Estimated Trajectory with Uncertainty. The binocular Good Features estimated trajectory is plotted with one-sigma uncertainty and the truth trajectory provided by the Vicon vision system.

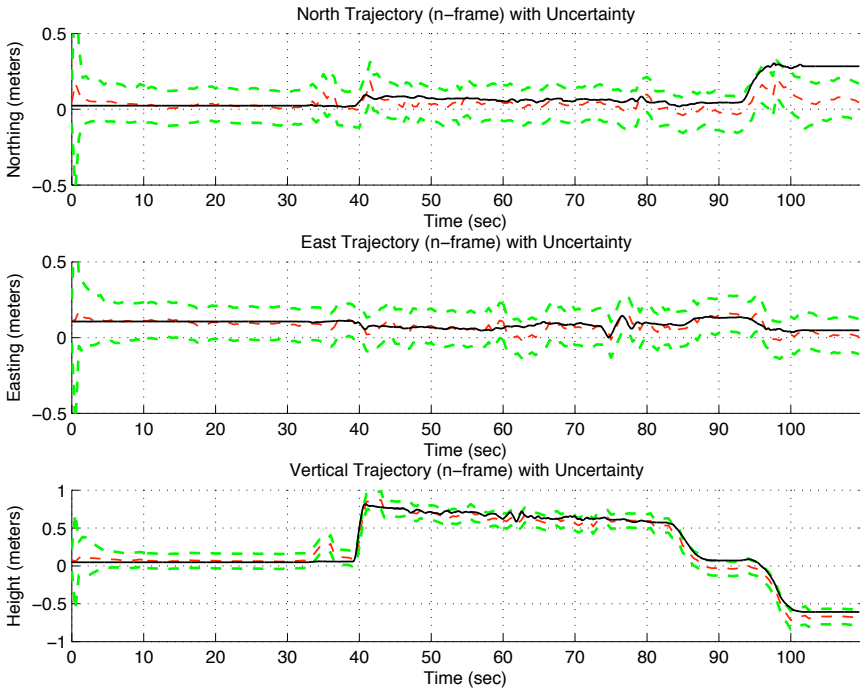



(a)

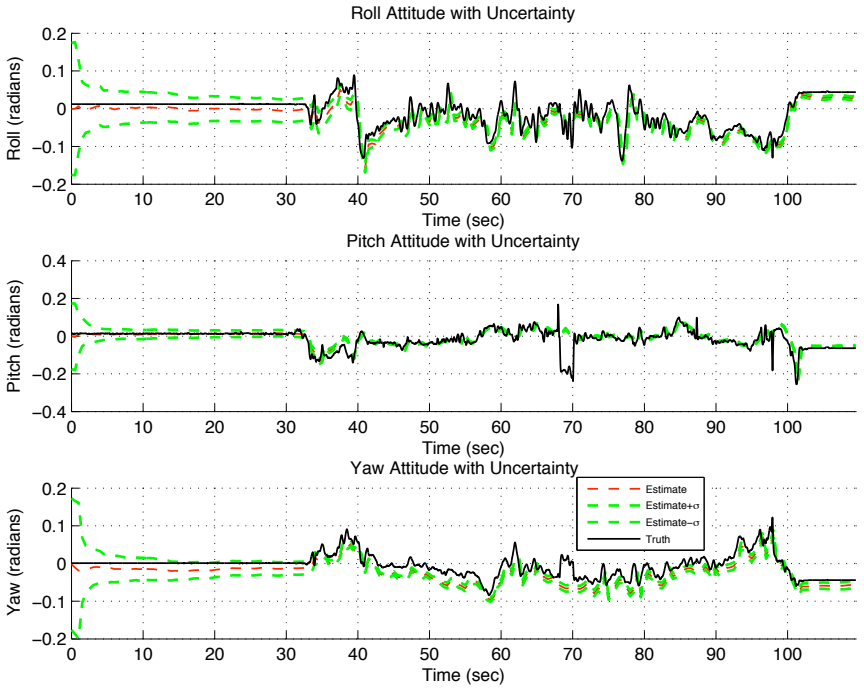



(b)

Figure 4.8: MAV Lab SIFT Monocular Estimated Trajectory with Uncertainty. The monocular SIFT estimated trajectory is plotted with one-sigma uncertainty and the truth trajectory provided by the Vicon vision system.




(a)




(b)

Figure 4.9: MAV Lab Good Features Monocular Estimated Trajectory with Uncertainty. The monocular Good Features estimated trajectory is plotted with one-sigma uncertainty and the truth trajectory provided by the Vicon vision system.

estimated trajectory had the best error performance, and the monocular Good Feature had the worst error due to the north position drift noted previously. All horizontal errors are constrained to 0.25 meters. The vertical estimated SIFT monocular and binocular trajectories performed slightly better than Good Features trajectories. All vertical errors were constrained to 0.2 meters.

Figure 4.12 shows RSS attitude errors of each estimated trajectory. As noted previously, the binocular Good Features estimated trajectory performed the worst because of the drift in the yaw dimension. The monocular estimated trajectories performed at least twice as good as the binocular versions. The could be explained by issues matching landmarks in the second camera where they were not initialized.

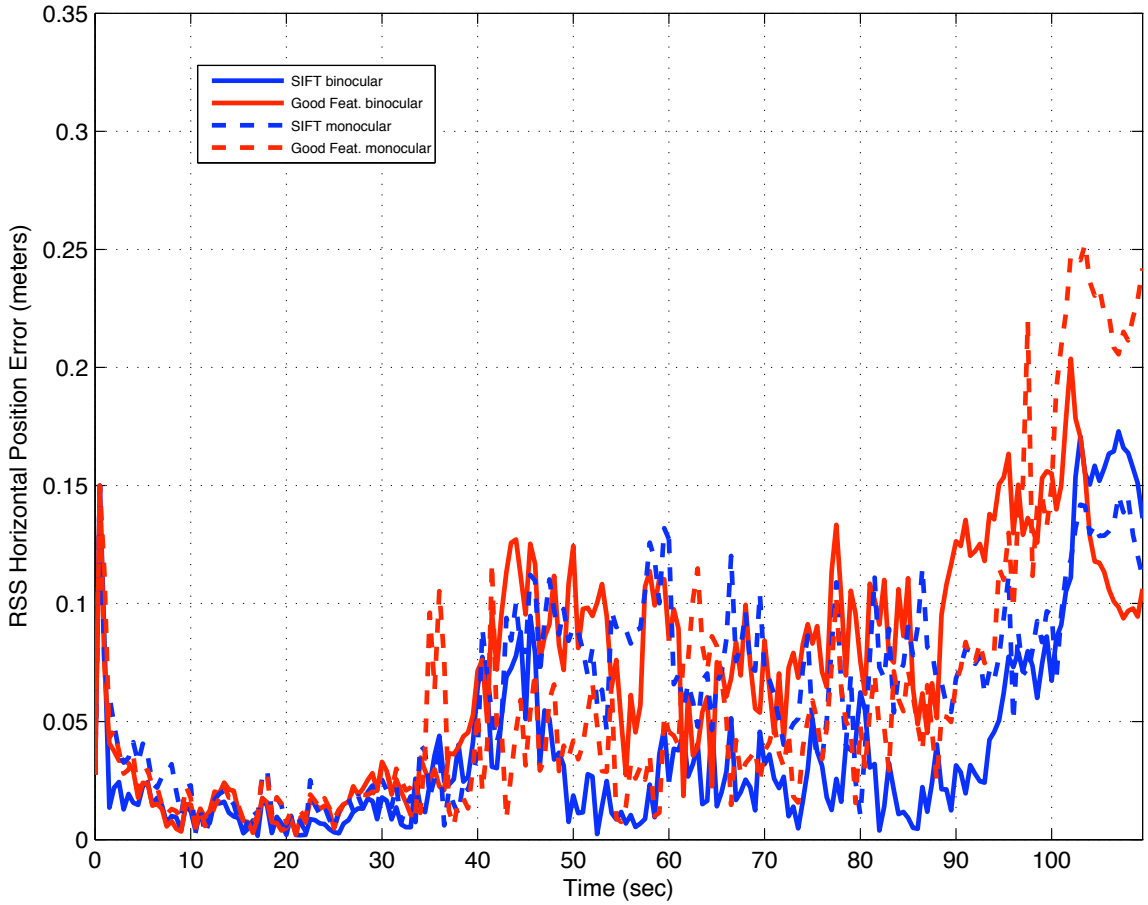



Figure 4.10: Root-Sum-Squared (RSS) Horizontal Position Error. Binocular and monocular estimated trajectory horizontal RSS error are compared for the SIFT and Good Features stochastic feature trackers.

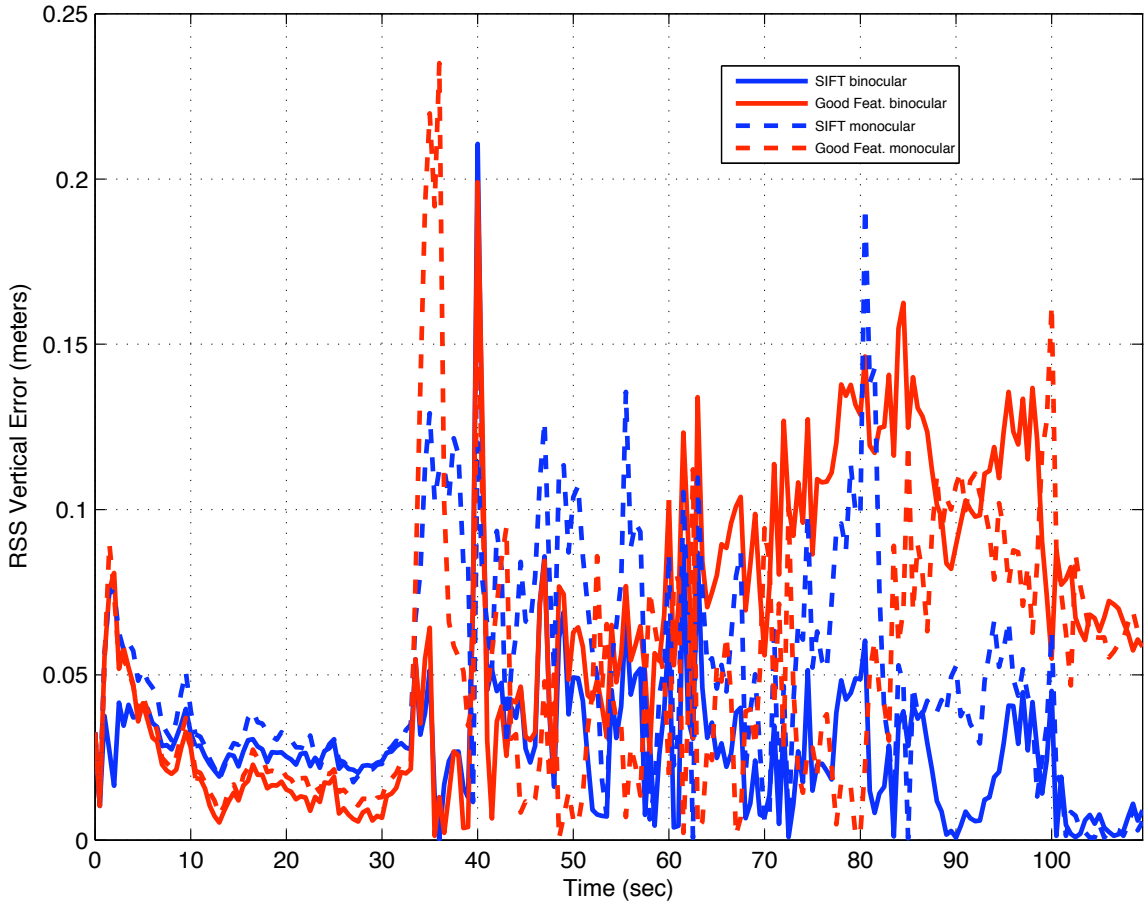



Figure 4.11: Root-Sum-Squared (RSS) Vertical Position Error. Binocular and monocular estimated trajectory vertical RSS error are compared for the SIFT and Good Features stochastic feature trackers.

Overall, these results show each stochastic tracker successfully estimates the truth trajectory during the hover condition. Yaw and north position were the only significant deviations from the truth trajectory in the binocular and monocular Good Features estimated trajectories.

This concludes the results for the indoor flight experiments conducted for this research. In the next chapter, conclusions from these results and recommendations for future work are presented.

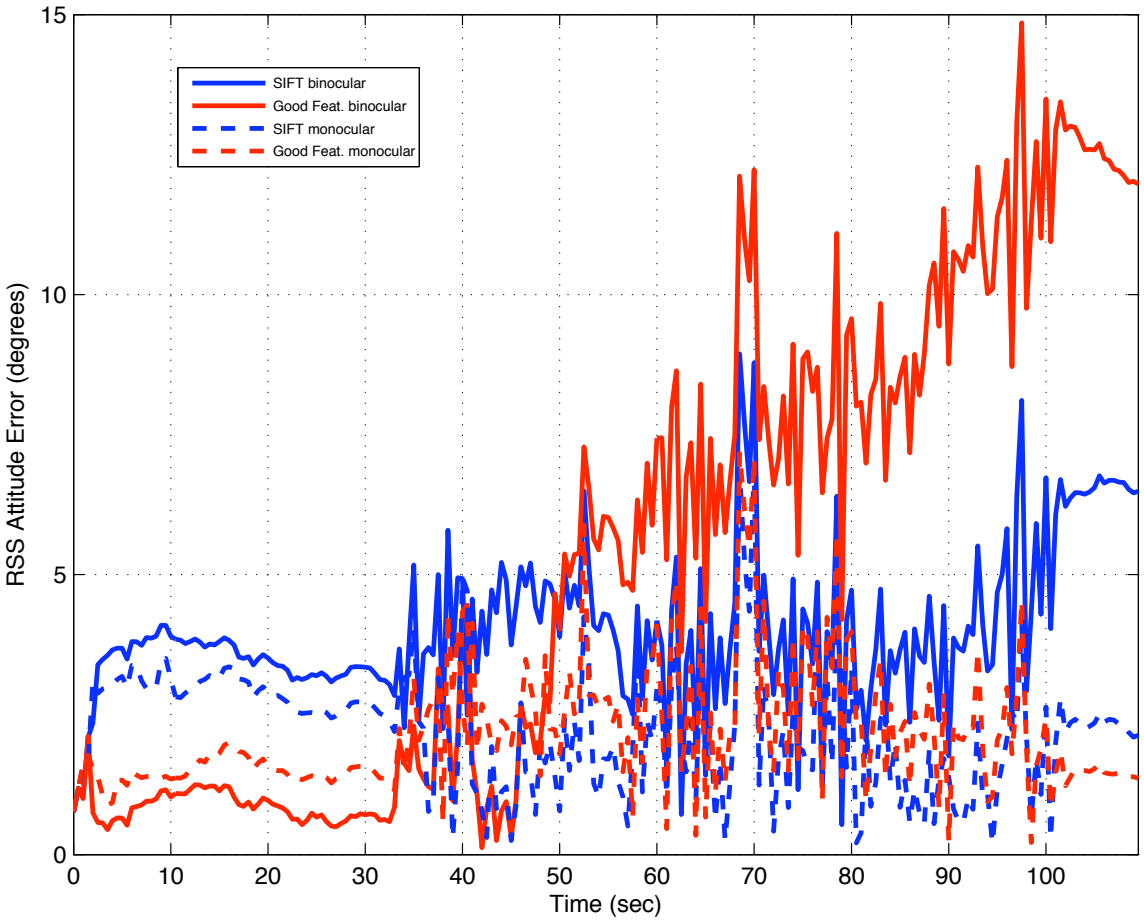



Figure 4.12: Root-Sum-Squared (RSS) Attitude Error. Binocular and monocular estimated trajectory attitude RSS error are compared for the SIFT and Good Features stochastic feature trackers.

# V. Conclusion

This research sought to develop a deeply integrated feature tracking algorithm that involved less computation than its predecessor. The previous algorithm used high-level features, while this research used low-level features with inertial aiding. Good Features extraction was selected for its repeatability and strength, two important characteristics of a feature detection algorithm. Results showed that the low-level transformation performed 12x faster and successfully reduced the overall computational complexity.

Rotational and 6DoF motion aiding were investigated to improve the low-level feature matching. Results showed that the deeply-integrated feature tracker was still faster than the predecessor, even with aiding techniques. With rotational aiding, a clear benefit was seen for severe flight trajectories. Six degree-of-freedom (6DoF) aiding was not entirely successful, but did show promising results in vertical trajectory aiding and reducing the number of landmarks initialized.

Three flight experiments showed that the low-level feature extraction can produced an accurate trajectory, on par with the previous robust features. The first experiment showed that with the stochastic constraint alone, the low-level feature extraction was able to constrain drift and produce an accurate trajectory. Aiding produced only subtle improvements for the first flight profile because of the lack of severe rotation. The second flight profile showed that aiding is necessary when severe attitude changes occur. Finally, the indoor MAV simulation showed that monocular low-level feature initialization produced a consistent estimated trajectory for an indoor hover condition. The only significant deviation from the truth occurred in the binocular and monocular Good Features stochastic tracker and was attributed to reflections on the surface of the MAV facility wall and less dominant horizontal trends in the image scene.

In either algorithm, the position estimate still drifted when features where poorly matched. This occurred in the first flight experiment were the image scene became saturated in one corner of the building. Without tracked features, errors

are introduced into the navigation solution that are unrecoverable without absolute reference updates.

A downside to using low-level features was the increase in the number of initialized features. Generally, thirty percent more features were initialized with the deeply-integrated feature tracker. This indicates that the SIFT-based feature tracking is more robust, but did not significantly affect the accuracy of the estimated trajectory or the speed of the deeply-integrated feature tracker.

## *5.1 Future Work*

The deeply-integrated feature tracker presented in this research demonstrated the capability of reducing the computational complexity by using a low-level feature transform with inertial aiding techniques. This section presents ideas for further improvements to speed and accuracy of the algorithm as well as alternative testing techniques.

Although not completed during this research phase, the low-level IAKF is being implemented on an indoor flying platform. With a proper debugging interface and recording capability, the real time operation of the filter could uncover timing issues or other real-time problems. Furthermore, the combination of the IAKF and a control algorithm has not yet been investigated. This final closed-loop test would validate the entire vehicles functionality and is the next step toward a fully autonomous vehicle.

If monocular vision is used during more general flight, further investigation of monocular landmark initialization will be necessary to produce accurate results. Features were assumed to have a mean depth with high uncertainty. The state vector is immediately augmented without measuring the depth. With low-level matching and significant movement, false matches will likely enter the filter and cause a corrupted estimate. A more appropriate landmark initialization using the stochastic constraint is introduced in [38]. This initialization calculates the uncertainty of a candidate feature and propagates the uncertainty into the next frame. A stochastically constrained

feature search is conducted and only after a successful match is the depth determined and the state vector augmented.

The truth reference provided by the MAV Lab at the Air Force Research Laboratory could help accurately characterize errors over an extended period. However, there are some disadvantages to using a visual marker system. First, the system produces a infrared flicker that could affect the vision system. Second, the size of the facility is limited, and the flight profile is constrained. Surveyed markers can provide an unconstrained position truth over a large area. However, this requires a debugging interface to indicate that a survey point has been reached. Also, the vehicle must pass over the surveyed points during the flight limiting the trajectory.

In this initial research, 6DoF motion aiding was limited to ceiling features. In this case, the normal vector is fully determined. As this research showed, additional processing time is available for more advanced techniques to determine the planar normal vector. Future research could investigate initializing the planar normal vector using image processing techniques. In addition, the vector could be continually estimated by augmenting the EKF's state vector.

This research selected Good Features detection and a image intensity descriptor. Other feature transformation combinations exist and warrant further investigation. The first recommended modification would use a gradient method for the low-level feature descriptor. The image gradient is readily available after the Good Features detection. After analyzing the gradient descriptor, other feature transformations could be introduced. The research in [24] [32] [33] provides an excellent starting point for feature transformation research.

Other nonlinear Kalman filtering techniques have been discussed during this development. Concurrent research at AFIT is investigating model-based mechanization for reducing drift in commercial inertial measurement units via a method of federated filtering. This could reduce the 10 second drift rate of the current IMU. The Unscented Kalman filter is another nonlinear Kalman filtering technique that trans-

forms sample points in the distribution through the nonlinear function. The multiple pose estimates from the Unscented filter could be passed to the feature descriptor aiding for weighting. Position and rotational observations are theoretically possible during such an update.

Finally, this research analyzed three flight profiles in detail to demonstrated the performance and accuracy of the deeply-integrated feature tracker. A Monte-Carlo analysis of repeated data collections over the same trajectory would give a better indication of the statistical performance of the filter.

## *5.2 Summary*

This research presented a deep integration of sensors necessary to reduce the computational requirements for small indoor flying vehicles. The method introduced, called the deeply-integrated feature tracker, used a low-level feature transform with inertial aiding of the descriptor. Results showed that the new tracker provided an accurate solution during multiple flight experiments. This filter is a key component to achieving a fully autonomous indoor flying vehicle in the very near future.

# *Bibliography*


1. Bay, Herbert, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. "Speeded-Up Robust Features (SURF)". *Computer Vision Image Understanding*, 110(3):346–359, 2008. ISSN 1077-3142.

2. Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features". *Proceedings of the Ninth European Conference on Computer Vision*. May 2006.

3. Bhanu, Bir, B. Roberts, and J. Ming. "Inertial navigation sensor integrated motion analysis for obstacle detection". *Proceedings IEEE International Conference on Robotics and Automation*, 954–959. May 1990.

4. Bouguet, J.-Y. "Pyramidal Implementation of the Lucas Kanade Feature Tracker", 1999. In OpenCV Documentation, Intel Corporation, Microprocessor Research Labs.

5. Bradski, Dr. Gary Rost and Adrian Kaehler. *Learning OpenCV, 1st edition*. O'Reilly Media, Inc., 2008. ISBN 9780596516130.

6. Brown, Duane C. "Close-Range Camera Calibration". *Proceedings of the Symposium on Close-Range Photogrammetry*, 855–866. January 1971.

7. Brown, Robert G. and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, New York, 1997. ISBN 9780471128397.

8. Derpanis, Konstantinos G. "The Harris Corner Detector", 2004.

9. Diel, David D., Paul DeBitetto, and Seth Teller. "Epipolar Constraints for Vision-Aided Inertial Navigation". *WACV-MOTION '05: Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05) - Volume 2*, 221–228. IEEE Computer Society, Washington, DC, USA, 2005. ISBN 0-7695-2271-8-2.

10. DMA WGS-84 Development Committee. *Department of Defense World Geodetic System 1984 - Its Definition and Relationships with Local Geodetic Systems*. Technical Report 8350.2, Defense Mapping Agency, Washington DC, Washington, DC, September 1987.

11. Engin, Z., M. Lim, and A.A. Bharath. "Gradient Field Correlation for Keypoint Correspondence". II: 481–484. 2007.

12. Harris, Chris and Mike Stephens. "A Combined Corner and Edge Detector". *The Fourth Alvey Vision Conference*, 147–151. 1988.

13. Hartley, Richard and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, March 2004. ISBN 0521540518.

14. Horn, Berthold K. P. "Projective Geometry Considered Harmful", 1999.

15. Ke, Yan and Rahul Sukthankar. "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors". *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, volume 2, 506–513. 2004.

16. Lewis, J. P. "Fast normalized cross-correlation". *Vision Interface*, 120–123. Canadian Image Processing and Pattern Recognition Society, 1995. URL `http://citeseer.ist.psu.edu/lewis95fast.html`.

17. Lowe, David G. "Object Recognition from Local Scale-Invariant Features". *Proc. of the International Conference on Computer Vision*, volume 2, 1150–1157. September 1999. Corfu, Greece.

18. Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints". *International Journal of Computer Vision*, 60(2):91–110, 2004.

19. Lucas, B. D. and T. Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision". *IJCAI81*, 674–679. 1981.

20. Ma, Xin, S. Sukkarieh, and Jong-Hyuk Kim. "Vehicle model aided inertial navigation". volume 2, 1004–1009 vol.2. 2003.

21. Maybeck, Peter S. *Stochastic Models Estimation and Control, Vol I.* Academic Press, Inc., Orlando, Florida 32887, 1979.

22. Maybeck, Peter S. *Stochastic Models Estimation and Control, Vol II.* Academic Press, Inc., Orlando, Florida 32887, 1979.

23. Microbotics, Inc. "MIDG 2 Specifications". Specification, January 2007. URL: http://www.microboticsinc.com/.

24. Mikolajczyk, K. and C. Schmid. "A performance evaluation of local descriptors". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

25. Nixon, Mark and Alberto S. Aguado. *Feature Extraction & Image Processing, Second Edition.* Academic Press, 2 edition, January 2008. ISBN 0123725380.

26. Noble, Allison. *Descriptions of Image Surfaces.* Ph.D. thesis, Department of Engineering Science, Oxford Univ., September 1989.

27. Government Accoutability Office, U. S. "Unmanned Aircraft Systems: Additional Actions Needed to Improve Management and Integration of DoD Efforts to Support Warfighter Needs". GAO-09-175, November 2008.

28. Office of the Secretary of Defense. "Unmanned Systems Roadmap (2007-2032)". U.S. Department of Defense, December 2007.

29. Overington, Ian. *Computer Vision: A Unified, Biologically-Inspired Approach.* Elsevier Science Inc., New York, NY, USA, 1992. ISBN 0444889728.

30. PixeLINK. "PixeLINK PL-A741 Machine Vision Camera Datasheet". Specification, April 2004. URL: http://www.pixelink.com/.

31. Qin, Lei, Wei Zeng, Wen Gao, and Weiqiang Wang. "Local invariant descriptor for image matching". *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, 2:ii/1025–ii/1028 Vol. 2, March 2005. ISSN 1520-6149.

32. Schmid, Cordelia, Roger Mohr, and Christian Bauckhage. "Comparing and evaluating interest points". *Computer Vision, 1998. Sixth International Conference on*, 230–235, January 1998.

33. Schmid, Cordelia, Roger Mohr, and Christian Bauckhage. "Evaluation of Interest Point Detectors". *International Journal of Computer Vision*, 37(2):151–172, 2000.

34. Shi, Jianbo and Carlo Tomasi. "Good Features to Track". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*. Seattle, June 1994.

35. Suter, D., T. Hamel, and R.-P. Mahony. "Visual Servo Control Using Homography Estimation for the Stabilization of an X4-flyer." *41st Conference on Decision and Control, CDC'02*. 2002.

36. Titterton, D.H. and J.L. Weston. *Strapdown Inertial Navigation Technology*. Peter Peregrinus Ltd., Lavenham, United Kingdom, 1997.

37. Trucco, Emanuele and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Upper Saddle River, New Jersey 07458, 1998.

38. Veth, Michael J. *Fusion of Imaging and Inertial Sensors for Navigation*. Ph.D. thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, September 2006.

39. Veth, Michael J. and John F. Raquet. "Fusion of Low-Cost Imaging and Inertial Sensors for Navigation". *Proceedings of the Institute of Navigation GNSS 2006*, 1093–1103. September 2006.

40. Vicon. "Vicon MX". Website, January 2009. URL: http://www.vicon.com/products/viconmx.html.

41. Zhu, Guopu, Shuqun Zhang, Xijun Chen, and Changhong Wang. "Efficient Illumination Insensitive Object Tracking by Normalized Gradient Matching". *Signal Processing Letters, IEEE*, 14(12):944–947, Dec. 2007. ISSN 1070-9908.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704–0188*



| 1. REPORT DATE *(DD–MM–YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From — To)* |
|---|---|---|
| 03–09–2009 | Master's Thesis | Sept 2007 — Mar 2009 |

**4. TITLE AND SUBTITLE**

Deeply-Integrated Feature Tracking
for
Embedded Navigation

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Jeffery R. Gray, 1Lt, USAF

**5d. PROJECT NUMBER**

JON 09-224

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT/GE/ENG/09-17

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Timothy J. Klausutis
Air Force Research Laboratory
101 W. Eglin Blvd, Bldg 13
Eglin AFB, FL 32542-6810
(850) 883-0887, timothy.klausutiseglin.af.mil)

**10. SPONSOR/MONITOR'S ACRONYM(S)**

AFRL/MNGI

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Approval for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

The Air Force Institute of Technology is investigating techniques to improve aircraft navigation using low-cost imaging and inertial sensors. Stationary features tracked within the image are used to improve the inertial navigation estimate. Features are tracked using a correspondence search between frames. Previous research investigated aiding these correspondence searches using inertial measurements. While this research demonstrated the benefits of further sensor integration, it still relied on robust feature descriptors to obtain a reliable correspondence match in the presence of rotation and scale changes. Unfortunately, these robust feature extraction algorithms are computationally intensive and require significant resources for real-time operation. Simpler feature extraction algorithms are much more efficient, but their feature descriptors are not invariant to scale, rotation, or affine warping which limits matching performance during arbitrary motion. This research uses inertial measurements to predict not only the location of the feature in the next image but also the feature descriptor, resulting in robust correspondence matching with low computational overhead.

**15. SUBJECT TERMS**

feature tracking, extended Kalman filter, vision-aiding, feature tracking, indoor navigation

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | LtCol Michael J. Veth |
| U | U | U | UU | 92 | 19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636, ext 4541; michael.vethafit.edu |